DAEWOO

MSX 2

대우퍼스컴 X-II

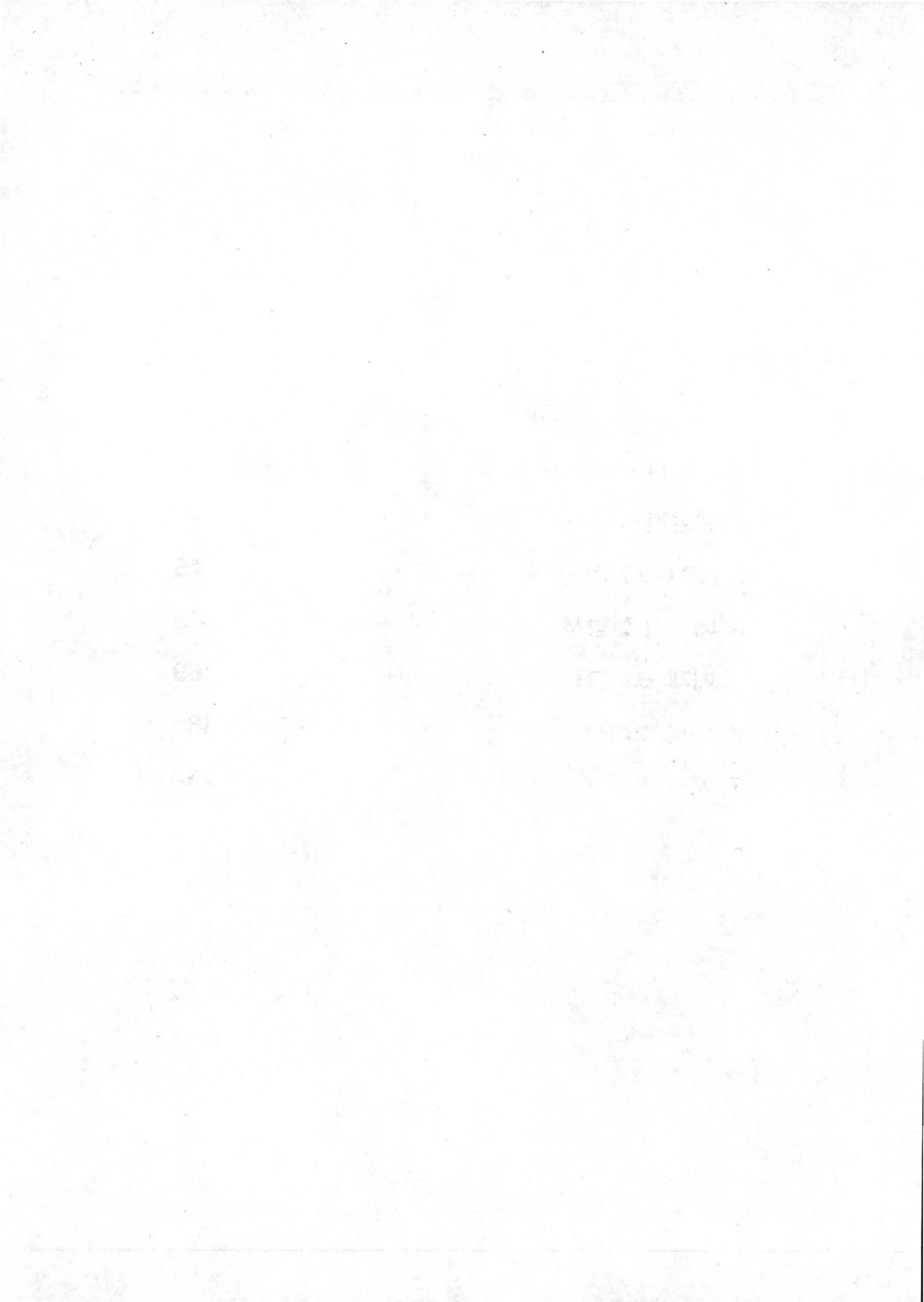
CPC-400/400S

머리말

저희 제품을 구입해 주셔서 감사합니다.
MSX 방식의 퍼스널 컴퓨터는 소프트웨어의 자유로운 호환성이 특징이며,
사용방법이 간편하여 누구나 쉽게 이용할 수 있읍니다.
MSX2 방식의 X-II는 기존의 MSX 방식인 아이큐 1000과 MSX2 방식인
아이큐 2000의 기능을 모두 포함하면서, 새로운 기능이 추가되었읍니다.
따라서 기존의 MSX 및 MSX2 방식의 소프트웨어를 모두 사용할 수 있으며,
3.5인치 FDD를 표준으로 내장하고 있어 많은 정보를 손쉽게 이용할 수 있읍니다.
또한 옵션으로 「수퍼임포즈보드」를 내장하여 「수퍼임포즈」 및 「디지타이즈」
기능을 수행할 수 있어 본격적인 컴퓨터 그래픽의 응용분야에 유용하게
사용하실 수 있읍니다.

이 책은 CPC-400 및 CPC-400S(수퍼임포즈보드가 내장된 모델)의
사용방법에 대하여
1. 설치편
2. 해설편
3. MSX-DOS란?
4. 베이직 설명서
5. 예제 프로그램
6. 수퍼 그래픽
7. 자료편
순으로 상세하게 설명하였읍니다.
이 책이 잘 활용되어 사용자 여러분에게 많은 도움이 되기를 바랍니다.

차 례

1

설치편

이 장에서는 취급 및 조작을 하기전의 기본적인 사항을 설명합니다.

1. 취급상의 주의사항

사용할 때에는 아래 사항에 유의하여 주십시오.

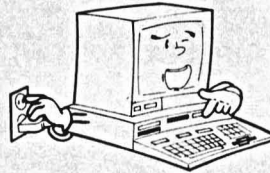
● 고온, 저온, 직사광선 및 먼지가 많은곳에 설치하지 마십시오.

● 텔레비젼 위에 본체를 올려 놓고 사용하지 마십시오.

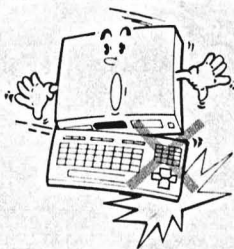
● 커피·쥬스 등의 음식물이나 꽃병의 물등을 흘리지 않도록 주의하십시오.

● 라디오등 수신기를 가까이 놓고 사용하면 수신기에 잡음이 생길 수도 있으므로 거리를 두고 사용하십시오.

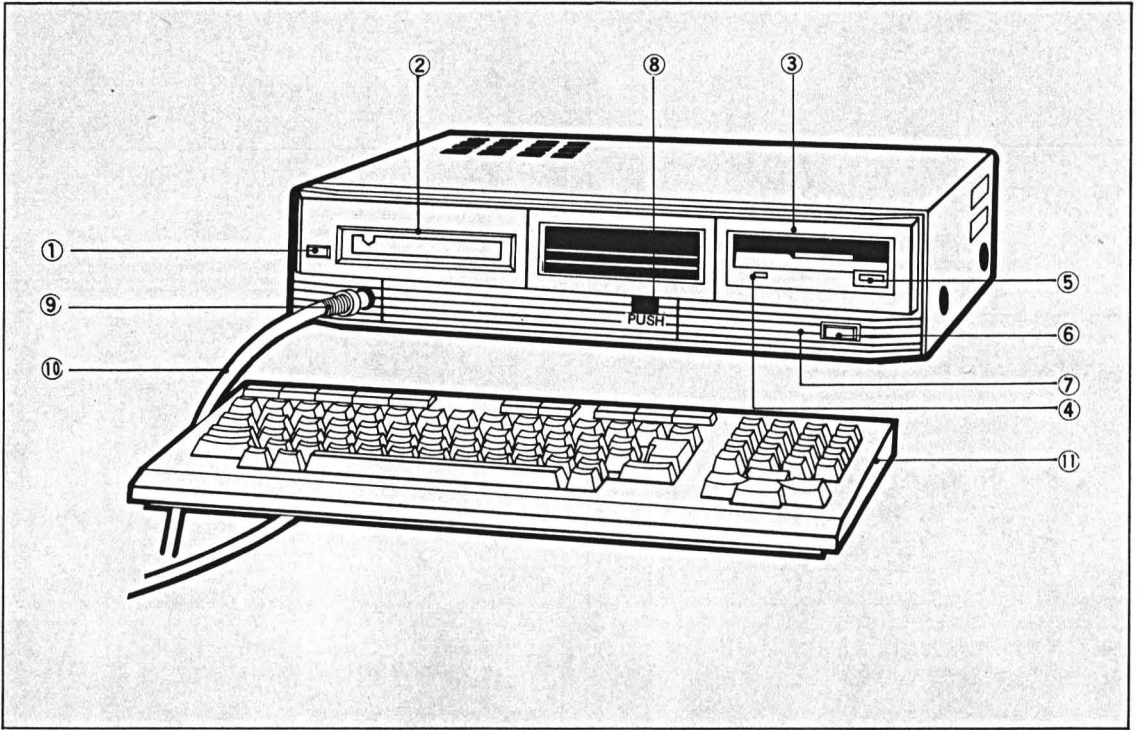
● 컴퓨터 위에 모니터 외에는 무거운 물건을 올려 놓지 마십시오.

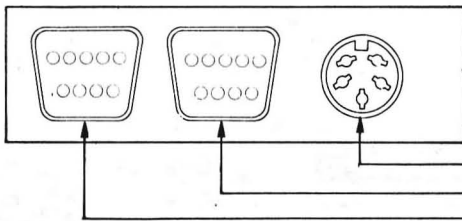
● 모니터, 프린터 등의 주변장치 등을 연결할 때는 반드시 전원을 끄고 해 주십시오.

● 본체를 세워서 사용하지 마십시오.

● 사용하지 않을 때는 전원 스위치를 끄고 전원 코드를 콘센트로 부터 뽑아 주십시오.

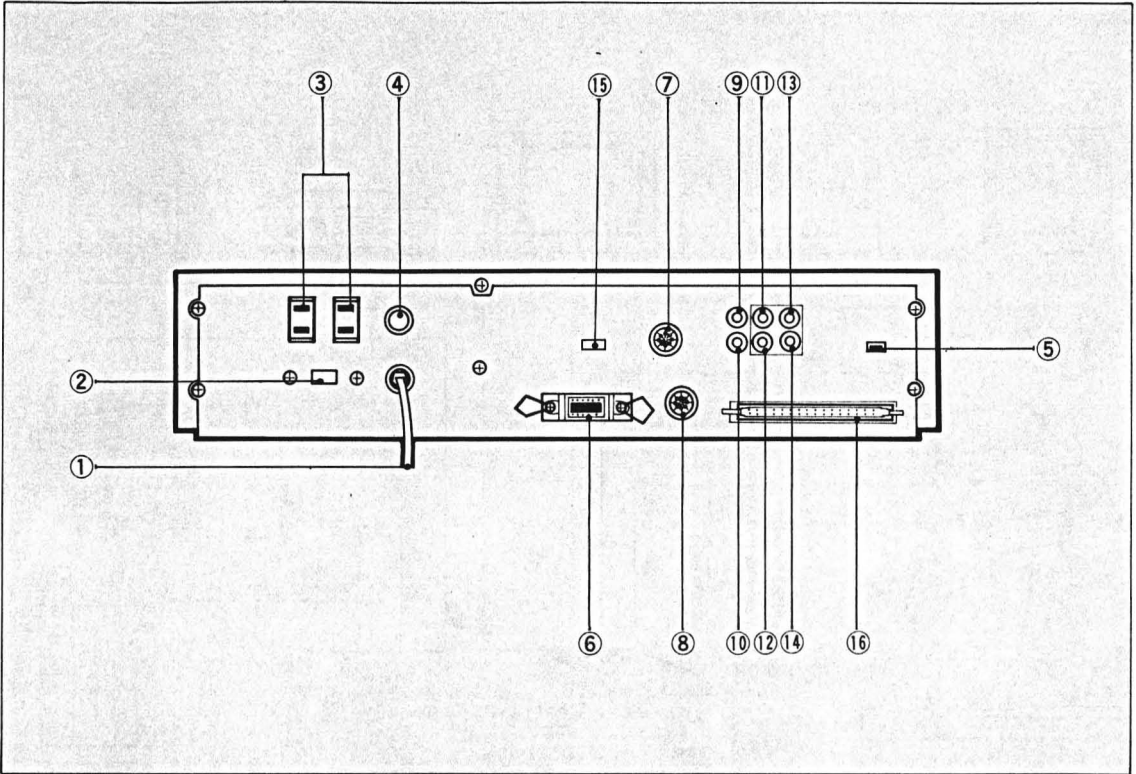
● 떨어 뜨리거나, 부딪히거나, 강한 충격을 주지 마십시오.

● 분해하지 마십시오.
내부를 손으로 만지면 위험하고, 고장의 원인이 될 수도 있읍니다.
만약 이상이 생겼을 때에는 대리점이나 서어비스 센터에 문의하여 주십시오.

2. 각부분의 명칭 및 기능

앞면

1. **리세트 스위치(RESET SWITCH) :** 눌렀다 떼면 지금까지 입력한 프로그램이 전부 지워지면서 시스템의 초기상태가 됩니다.
2. **슬로트(SLOT) :** 카트리지(CARTRIDGE)또는 주변기기의 연결에 사용합니다.
3. **디스크 드라이브 A :** 디스켓을 넣을 수 있읍니다.
4. **드라이브A의 동작표시 램프 :** 드라이브A가 동작중일 때 램프에 불이 들어옵니다.
5. **드라이브A의 이젝트 버튼 :** 버튼을 누르면 안에 있던 디스켓이 밖으로 나옵니다.
6. **전원스위치 :** 버튼을 누르면 컴퓨터에 전원이 들어옵니다.
7. **전원램프 :** 컴퓨터의 전원을 켜면, 램프에 불이 들어옵니다.
8. **도아-오픈(Door-open) :** 「PUSH」라고 쓰여진 부분을 누르면 케이스가 열려지면서 아래 그림과 같이 됩니다.

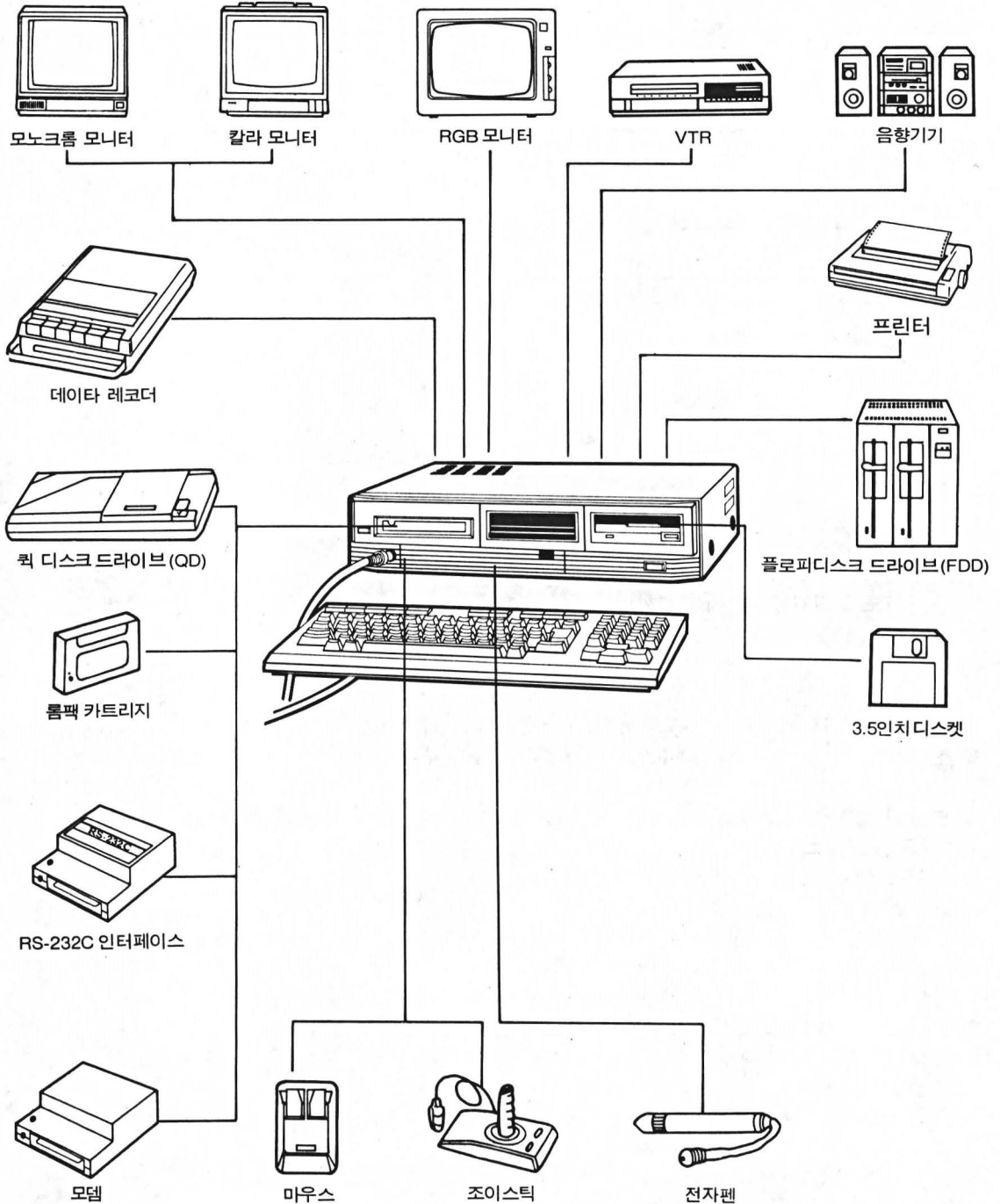
도어의 내부 :

9. **키보드 접속단자 :** 키보드를 연결합니다.
10. **키보드 연결 케이블 :** 키보드를 컴퓨터에 연결해주는 케이블
11. **키보드 :** 프로그램을 입력할 때 사용합니다.

※ 디스크드라이브를 1대 더 연결하려면 전문점이나 서어비스센터에 문의하여 주십시오.

뒷면

1. 전원코드 : 전원콘센트에 연결합니다.
2. 전원선택 스위치 : 100V (100V) 또는 220V (220V) 의 전원을 선택할 수 있읍니다. 처음에는 220V 로 조정되어 있읍니다.
3. 보조전원 콘센트 : 공급된 전원이 나옵니다.
4. 휴즈(FUSE) : 컴퓨터에 과도한 전류가 흐를때 컴퓨터를 보호하기 위해 휴즈가 파괴됩니다.
5. 스트레오/모노 스위치 : 컴퓨터의 음성출력을 모노로 할 것인지 아니면 스트레오로 할 것인지를 선택합니다.
6. 프린터 출력단자 : MSX 표준방식(준 센트로닉스 방식)의 프린터를 연결합니다.
7. RGB 영상출력단자 : 아날로그 입력 RGB 모니터를 연결합니다.
8. 카세트 단자 : 데이타 레코더를 연결합니다.
9. 영상입력단자 : 외부 영상 신호의 입력에 사용합니다. 수퍼임포즈나 디지타이즈 기능을 사용할 때 사용합니다.
10. 영상출력단자 : 영상신호의 출력에 사용합니다.
11. 음성입력단자(R) : 외부 음성신호의 입력에 사용합니다. (오른쪽 채널)
12. 음성출력단자(R) : 음성 신호의 출력에 사용합니다. (오른쪽 채널)
13. 음성입력단자(L) : 외부 음성신호의 입력에 사용합니다. (왼쪽 채널)
14. 음성출력단자(L) : 음성신호의 출력에 사용합니다. (왼쪽 채널)
15. 칼라/모노 스위치 : 칼라 모니터를 사용할 것인가, 모노크롬 모니터를 사용할 것인가를 선택합니다.
16. 확장버스(EXPANSION BUS) : 주변장치의 연결에 사용합니다.

* 수퍼임포즈 기능이 없는 컴퓨터는 ⑨⑪⑬ 단자가 없읍니다.

3. 각부분의 연결 및 설치

1. 주변기기와의 연결

1) 데이타 레코더(DPR-600)

컴퓨터의 외부기억 장치로서 프로그램을 테이프에 보존합니다.

2) 퀵 디스크 드라이브(QD:DPQ-280)

컴퓨터의 외부기억 장치로서 프로그램을 디스켓(2.8인치)에 보존합니다.

3) 롬팩 카트리지

롬팩에 있는 프로그램을 사용할 수 있읍니다.

4) RS-232C 인터페이스(CMC-01)

RS-232C 규격의 인터페이스(INTERFACE)를 연결할 수 있읍니다.
직렬 데이타(SERIAL DATA) 전송을 위한 RS-232C는 직접 컴퓨터끼리 통신할 수 있으며, 모뎀을 이용하면 전화선을 통한 컴퓨터끼리의 통신에도 이용됩니다.

5) 모뎀

MSX 전용 모뎀 카트리지를 연결할 수 있읍니다.

6) 마우스

조이스틱 단자에 연결하여 사용할 수 있으며, 그림을 그릴 때에 편리하게 사용할 수 있는 입력장치입니다.

7) 조이스틱(CPJ-902)

조이스틱 단자에 연결하여 사용할 수 있으며, 게임을 할때에 사용합니다.

8) 전자펜

입력장치로서 화면의 좌표를 컴퓨터에 입력시킵니다. 이것을 이용하여 화면에 직접 그림을 그릴 수 있읍니다.

9) 3.5인치 디스켓

많은 양의 데이타를 저장할 수 있는 외부기억 매체로서, 배밀도의 경우 포맽을 하면 737280 바이트의 데이타를 저장할 수 있읍니다.

10) 플로피 디스크 드라이브

본체에 내장된 3.5인치 디스크 드라이브 외에 본체 뒷면의 확정버스에 5¼인치 혹은 3.5인치 플로피 디스크 드라이브를 추가로 더 연결할 수 있읍니다.
이 경우 본체의 드라이브명은 "A >", "B >"로 표시되고, 뒷면의 확장버스에 연결된드라이브의 드라이브명은 "C >", "D >"로 표시됩니다.

11) 프린터

MSX 전용프린터(DPP-1400A, CPP-1200등) 를 사용할 수 있읍니다.

12) 음향기기, VTR

수퍼임포즈 보드가 내장된 시스템에서는 음향기기 및 비디오기기(VTR, 비디오 카메라등) 등을 연결하여 AVC(Audio Visual Control) 시스템을 구성할 수 있읍니다.

13) 디스플레이 장치

X-II에서는 콤포지트 영상신호 및 아날로그RGB 신호가 나오므로 다양한 디스플레이 장치를 사용할 수 있읍니다. 특히 강력한 기능의 고해상도 칼라그래픽을 이용하려면 아날로그RGB 모니터를 사용하면 좋습니다.
한편 인터레이스 모드에서는 40×24자의 한글 및 32×24자의 한자를 사용할 수 있으므로 업무용으로 사용하시기에 좋습니다(이경우 인터레이스가 가능한 전용모니터를 이용하십시오).
디스플레이 장치를 연결하시기 전에는 반드시 칼라/모노 선택스위치를 확인해 주십시오.
보다 더 상세한 연결방법은 각 주변장치의 사용설명서를 참조하여 주십시오.

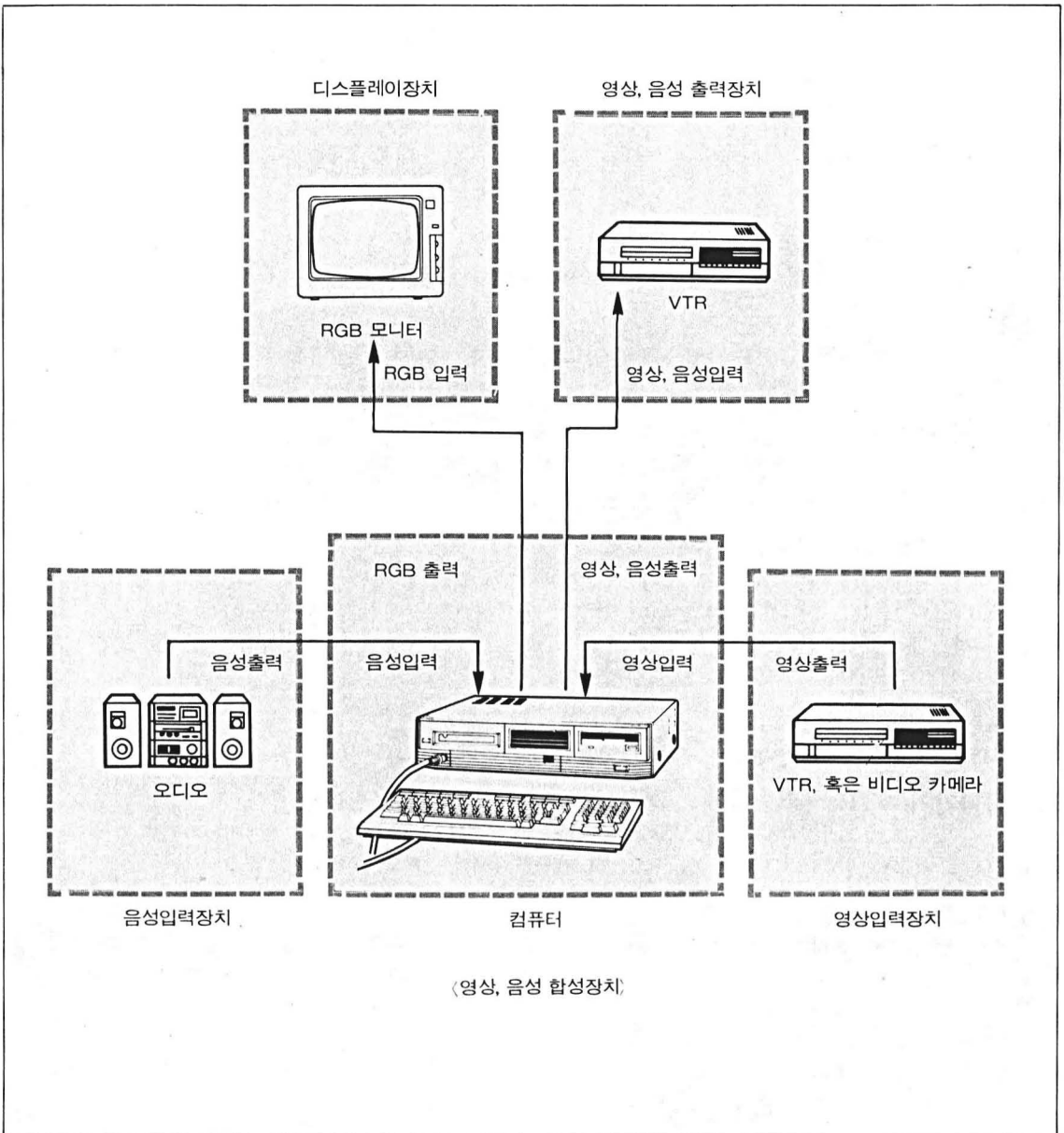
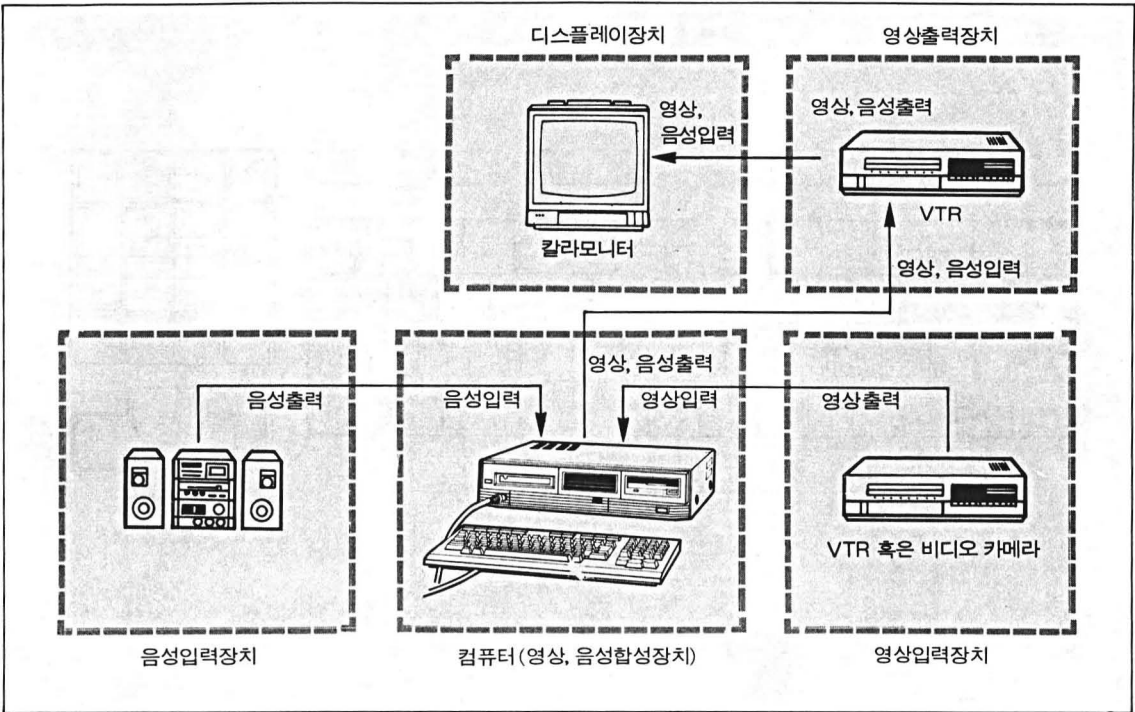
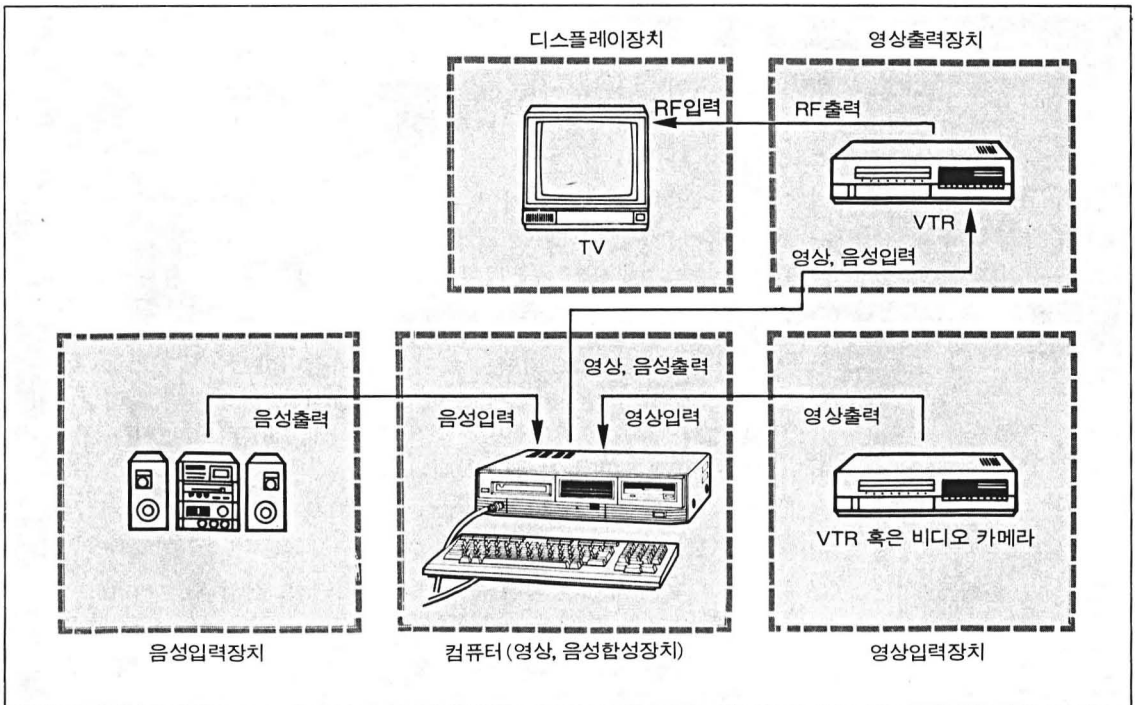
2. AVC(Audio Visual Control)시스템의 구성

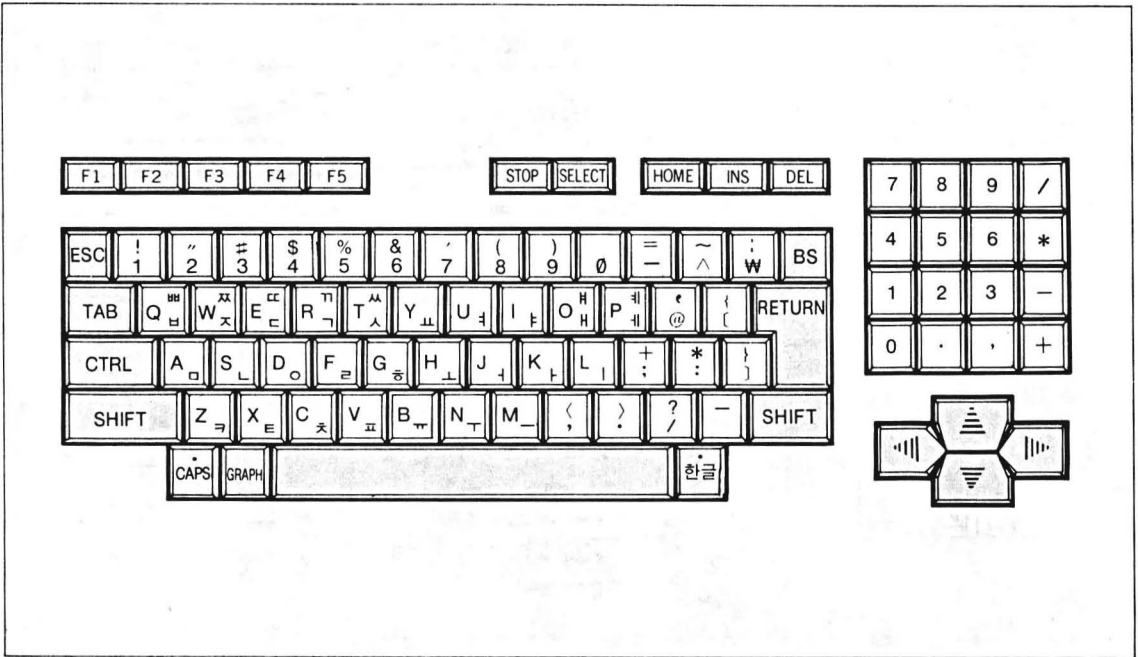
수퍼임포즈 보드가 내장된 시스템에서는 각종 음향기기나 비디오기기(VTR, 비디오 카메라등)를 연결하여 AVC 시스템을 구성할 수 있읍니다. 즉 외부의 음성신호와 영상신호를 컴퓨터의 신호와 합성을 하여, 외부에 출력할 수 있읍니다.
이와같은 기능을 이용하여 비디오 편집이나 다른 특수한 용도로 쓸 수 있읍니다. 보다 더 자세한 내용은 해설편의 「수퍼임포즈, 디지타이즈」항을 참조해 주십시오.
아래의 그림들은 AVC 시스템을 구성하는 몇가지 예를 들었읍니다.

RGB 모니터를 이용하는 경우

〈영상, 음성 합성장치〉

칼라 모니터를 이용하는 경우

디스플레이장치
영상출력장치
영상,
음성입력
영상, 음성출력
칼라모니터
VTR
영상, 음성입력
영상, 음성출력
음성출력
음성입력
영상입력
영상출력
VTR 혹은 비디오 카메라
음성입력장치
컴퓨터(영상, 음성합성장치)
영상입력장치

TV를 이용하는 경우

4. 키보드(KEY BOARD)

키 보드는 아래와 같이 배열되어 있읍니다.

특수키

SHIFT	**쉬프트.**	영 대문자(단, CAPS 에 불이 켜져 있을 때는 영 소문자) 또는 한글, 숫자, 부호키 등의 상단의 문자를 표시할 때 사용합니다. 이 키는 키 보드의 좌우 양쪽에 있으며, 어느 쪽을 사용해도 좋읍니다.
CAPS	**캡스.**	영 대문자를 표시할 때 사용합니다. 한 번 누르면, 영 대문자 상태로 되면서 램프에 불이 켜집니다. 다시 한번 누르면, 불이 꺼지면서 영 소문자 상태로 됩니다. 또, 영 대문자 상태에서 SHFT 와 함께 영 문자를 치면 영 소문자가 표시됩니다.
GRAPH	**그래픽.**	이 키를 누르면서 문자키를 치면, 그래픽 기호가 표시됩니다.
CTRL	**컨트롤.**	이 키를 누르면서 문자키를 치면 특수한 기능을 실행합니다. 각각의 기능에 대해서는 자료편의 「제어코드」항을 참조하십시오.
(커서키)	**커서.**	각각 커서를 상하좌우로 이동시킬 때 사용합니다.
INS	**인서트.**	이 키를 치면 삽입모드로 되면서 커서가 작아집니다. 이때, 문자를 치면 커서 자리에 문자가 삽입되면서, 뒤에 있는 문자를 한 문자씩 밀려납니다. 삽입모드에서 다시 이 키를 치던가, 커서키 또는 RETURN 키를 치면, 삽입모드가 해제되면서 커서도 원래의 형태로 되돌아옵니다.
DEL	**딜리트.**	현재 커서가 있는 곳의 문자가 삭제되면서, 커서 우측에 있는 문자는 모두 왼쪽으로 한문자씩 이동됩니다.
BS	**백스페이스.**	현재 커서가 있는 바로 앞의 한문자가 삭제되면서, 커서 우측에 있는 문자는 모두가 왼쪽으로 한문자씩 이동됩니다.
TAB	**탭.**	이 키를 누르면 커서는 다음 탭 위치로 이동되고, 그 사이는 공백으로 됩니다. 탭 위치는 1, 9, 17 등 1부터 8 자씩 더해갑니다.

HOME / CLS **홈.** 커서가 화면의 좌측 상단으로 이동합니다. SHIFT 키를 누르면서 이 키를 치면, 커서가 화면의 좌측상단으로 이동함과 동시에 화면이 모두 지워집니다.

STOP **스톱.** 프로그램 실행 중 이 키를 치면, 프로그램이 일시정지 상태로 되며, 다시 이 키를 치면 실행이 계속됩니다. 정지 상태에서는 다른 키를 입력할 수 없습니다. 키를 입력하고 싶을 경우는 CTRL 을 누르면서 STOP 을 치십시오. 이 때, 처리를 계속하려면, C, O, N, T, RETURN 을 칩니다.

RETURN **리턴.** 키 보드로부터 입력된 문자를 컴퓨터에 기억시킬 때 사용합니다.

F1 ~ F5 **펑션.** 각 키에 정의된 문자열이 입력됩니다. SHIFT 와 함께 누르면 F6 ~ F10 이 동작됩니다.

SELECT **셀렉트.** 프로그램중에서 이용합니다. 제어코드 24를 입력합니다.

ESC **이스케이프.** 프로그램중에서 이용합니다. 제어코드 27를 입력합니다.

한글 이 키를 치면 한글모드로 되어, 문자키를 치면 한글의 자모가 표시됩니다
다시 한글 키를 치면, 한글모드가 해제됩니다.
한글모드로 되면 램프에 불이 켜집니다.

문자키

각각의 문자키는 특수키와 조합함으로써 여러 종류의 문자를 나타낼 수 있습니다.
여기서 「+」와 「,」는 다음과 같은 의미를 가집니다.
A + B —— A 키를 누른 상태에서 B 키를 칩니다.
A, B —— A 키를 친 다음 B 키를 칩니다.

누른 키	표시된 문자
P	p (영 소문자)
SHIFT + P	P (영 대문자)
GRAPH + P	π (도형문자)
CAPS, P	P (영 대문자)
CAPS, SHIFT + P	p (영 소문자)
한글, P	ㅔ(한글)
한글, SHIFT + P	ㅖ(한글)

누른 키	표시된 문자
5	5
SHIFT + 5	%

누른 키	표시된 문자
:	:
SHIFT + :	*

숫자키

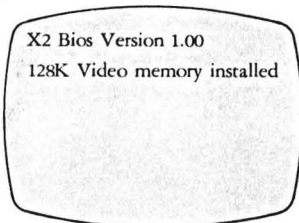
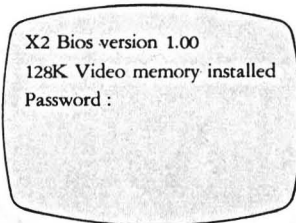
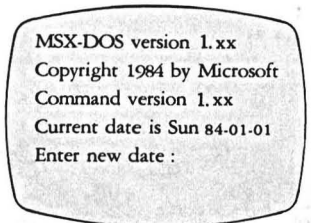
7	8	9	/
4	5	6	*
1	2	3	—
0	.	,	+

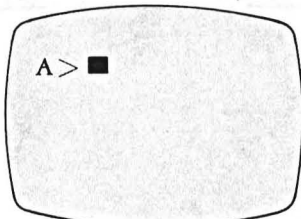
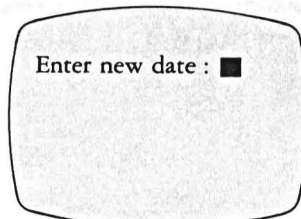
숫자와 연산기호로 구성되어 있어서, 계산기처럼 사용할 수 있습니다.

5. 컴퓨터의 동작

주의 X-Ⅱ는 처음에는 전압 220V로 조정되어 있읍니다. 100V에서 사용할 때에는, 뒷면에 있는 전압선택스위치를 100V로 조정한 뒤 사용하십시오.

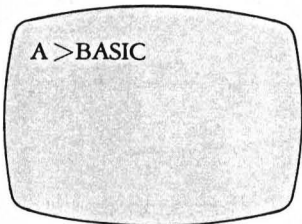
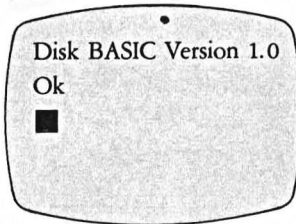
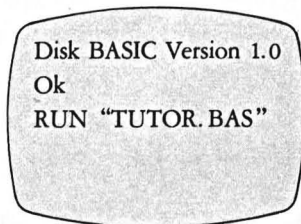
1. 컴퓨터의 동작

먼저 주변기기의 연결상태를 확인한 후 주변기기의 전원을 켭니다. 다음 본체의 디스크 드라이브에 디스켓을 넣은 후 뒷장의 2.「디스켓에 대하여」를 참조하여 주십시오. 본체의 우측하단에 있는 전원스위치를 누르면 아래 〈그림 1〉의 화면이 나타납니다.
이때 만약 암호(Password)가 설정되어 있었다면 〈그림2〉와 같이 「Password:」라는 메시지가 나온 뒤 암호의 입력을 기다립니다.(Set Password 명령을 참조하여 주십시오.)
만약 암호를 잊어버린 경우에는 전원을 켜면서 [GRAPH]와 [STOP] 키를 동시에 누르면 〈그림3〉의 화면으로 넘어갑니다.

(그림 1) (그림 2) (그림 3)

주의 X-Ⅱ는 MSX2 방식으로서 이전의 MSX 소프트웨어를 모두 사용할 수 있으나, MSX2의 표준한글코드는 2바이트 코드이기 때문에, 이전의 MSX에서 한글코드(n바이트 한글코드)를 사용한 소프트웨어를 MSX2에서 사용하고자 할 때는 〈그림 3〉의 화면이 나오기 이전에 [ESC] 키와 [SELECT] 키를 동시에 누르면 됩니다. 또, 베이직 프로그램으로 한글모드의 초기치를 지정할 수도 있읍니다.
(SET SYSTEM, SET HAN 명령을 참조하여 주십시오.)

〈그림3〉의 상태에서 날짜를 변경하고 싶으면 〈년〉-〈월〉-〈일〉의 순서로 날짜를 입력한 뒤 [RETURN] 키를 칩니다. 만약 날짜를 입력하지 않고 [RETURN] 키를 누르면 이전의 날짜가 기억되며, 〈그림4〉와 같이 「A >」라는 프롬프트가 나옵니다.

(그림 4) (그림 5)

이와같은 상태를 MSX-DOS 상태라고 합니다. MSX-DOS란 MSX용의 DOS(Disk Operating System)로서 MS-DOS와 데이타 화일의 호환성이 있으며, CP/M과 펑션 콜(Function Call)수준에서 호환성이 있읍니다. (더 자세한 사항은 「MSX-DOS란?」편을 참조하십시오.) 만약 시스템 디스켓을 넣지 않고 전원을 켠 경우에는 〈그림5〉와 같은 화면이 나타납니다. 날짜 변경요령은 시스템 디스켓이 있는 경우와 같습니다. 만약 〈그림3〉이 화면에 나오기 이전에 [SHIFT] 키를 누르면 다음과 같이 됩니다.

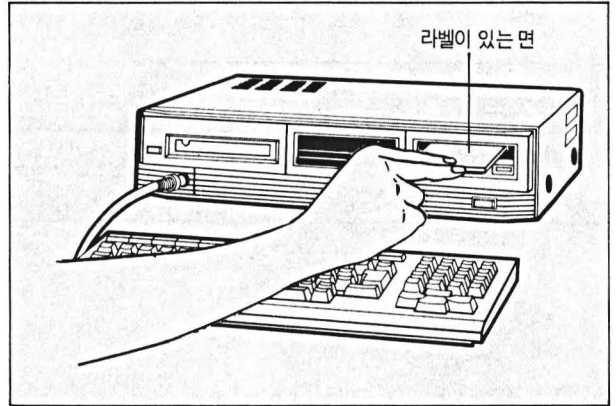
	사용가능한 디스크 드라이브	베이직 상태에서 사용가능한 사용자 메모리
[SHIFT] 키를 누른 경우	「A, B 드라이브」모두 사용불가	28815 바이트
키를 누르지 않은 경우	모두 사용 가능	23431 바이트

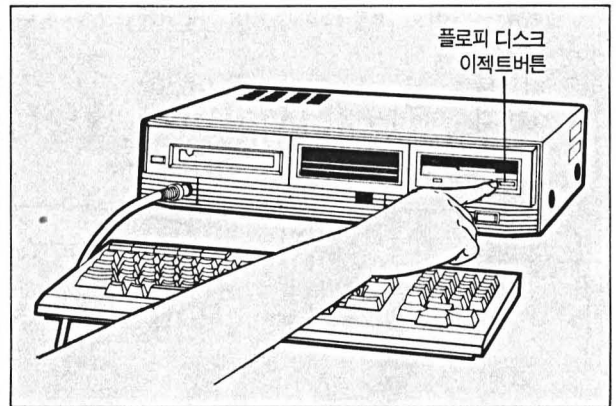
MSX-DOS나 디스크 베이직을 사용하고 싶지 않을 경우에는 전원을 키면서 [SHIFT] 키를 누르면 됩니다. 그러면 마치 디스크 드라이브가 설치되어 있지 않은 것과 같은 상태가 됩니다.
만일 〈그림4〉와 같은 MSX-DOS 상태에서 디스크 베이직을 사용하고 싶을 경우 〈그림6〉과 같이 「A > BASIC」이라고 입력한 뒤 [RETURN] 키를 칩니다. 그러면 〈그림7〉과 같이 디스크 베이직의 명령대기상태가 됩니다.
만일 컴퓨터를 처음 사용하시는 분이라면 〈그림8〉과 같이 입력한뒤 [RETURN] 키를 치면 기본적인 컴퓨터 사용법을 쉽게 배우실 수 있읍니다.

(그림 6) (그림 7) (그림 8)

2. 디스켓에 대하여

디스켓이란 많은 양의 정보(Data)를 보관하기 위한 보조기억 매체입니다. 디스켓에 대한 보다 더 자세한 설명은 해설편의 「플로피 디스크」편을 참조하여 주십시오.

■ 디스켓을 넣는 방법

반드시 라벨(Lavel)이 있는 면을 위로하여 손으로 끝까지 밀어 넣으시면 됩니다.
(다 들어가면 자동적으로 닫힙니다.)

■ 디스켓을 빼는 방법

디스크 드라이브의 우측 하단에 있는 버튼을 누르면 자동적으로 디스켓이 나옵니다.

주의 디스크 드라이브의 표시등에 불이 켜져있을 때(「Save」 및 「Load」중일 때)는 디스켓을 집어넣거나 빼지 마십시오. 디스켓에 저장된 정보가 파괴되는 경우도 있읍니다.

6. 컴퓨터에 이상이 있을 경우

X-II를 사용하시다가 이상이 생기면, 구입하신 대리점이나 서어비스센터에 문의하시기 전에 먼저 다음과 같은 사항들을 확인해 주십시오.

1. 전원을 켰는데 화면이 나타나지 않을 경우

*모니터의 전원이 켜 있는지 확인합니다.
*모니터의 콘트라스트(Contrast)와 명암조절이 잘 되어 있는가를 확인합니다.

2. 전원을 켰을 때, 화면에 노이즈(Noise)만 나타날 경우

*디스플레이 장치와 컴퓨터의 연결이 바르게 되어 있나 확인합니다.
*컴퓨터의 전원이 제대로 연결되어 있나 확인합니다. (전원램프에 표시됩니다)
*카트리지를 사용할 때는 카트리지가 잘 꽂혀져 있는지 확인합니다.

3. 화면이 선명하지 못하고, 화면이 흐르거나 노이즈가 많을 경우

*디스플레이의 화면조정 손잡이로 좋은 화질을 얻도록 합니다.
*화면이 흐를 때는 디스플레이장치의 수평 및 수직동기 손잡이를 조절합니다.
*컴퓨터를 설치한 장소가 외부의 간섭신호를 받기 쉬운 곳은 아닌지 확인합니다.
*컴퓨터는 다른 기기에서의 간섭신호에 영향을 받기도 하지만, 컴퓨터 자체에서도 약간의 방해신호가 발생하므로, 되도록이면 TV나 라디오를 컴퓨터와 동시에 사용하지 않는 것이 좋습니다.

4. 소리가 나지 않을 경우

*음성케이블의 연결이 잘되어 있는지를 확인하고, 모니터의 음량조절 손잡이를 조절합니다.

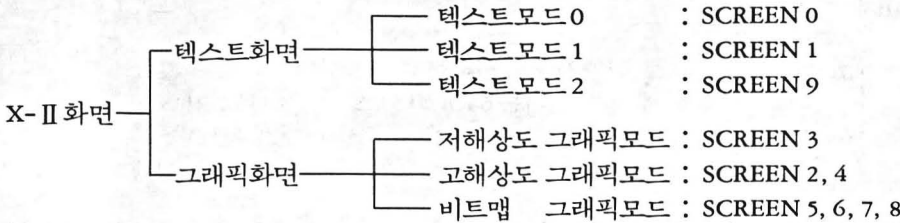
5. 화면에 아무런 반응이 없을 경우

*컴퓨터 사용중에 커서가 화면에서 사라지고, 키를 눌러도 화면에 표시되지 않는 경우, [CTRL] + [STOP] 키를 눌러서 커서가 화면에 나타나는가를 조사합니다. 화면에 커서가 나타났는데도 키 입력이 안되면, [RESET] 키를 눌러 다시 시작합니다.
그래도 컴퓨터가 이상하면, 아프터 서어비스를 의뢰하십시오.
*한 개의 전원콘센트에 전열기와 같이 전력소모가 큰 전기기구와 함께 접속해서 사용하면, 전기기구의 전원상태에 따라서 컴퓨터의 동작이 불안정해질 수 있으므로, 되도록이면 전력소모가 큰 전기기구와는 같이 사용하지 않도록 하십시오.

2
해설편

이 장에서는 X-Ⅱ가 가지고 있는 특수한 기능들에 관하여 설명합니다.
설명중 나오는 몇가지 BASIC명령에 관해서는 BASIC설명서를 참조하십시오.

1. 화면모드

X-Ⅱ의 화면에는 다음과 같은 종류가 있습니다.

텍스트 화면은 주로 문자를 표시하기 위한 화면입니다.
그래픽 화면은 그림이나 도형을 그리기 위한 화면이며, 몇 개의 점들을 한 단위로 하여 색을 칠하는 저, 고해상도 그래픽 모드와 화면의 각 점마다 색을 칠할 수 있는 강력한 그래픽 기능을 가진 비트맵 그래픽 모드가 있습니다.
한편, 각 화면모드에 따라서 아래와 같은 다양한 기능을 가지고 있습니다.

● **색상**

512색 중에서 16색을 선택하여 화면에 표시하거나, 256색을 동시에 표시할 수 있습니다.

● **스프라이트(動画)**

움직이는 그림을 편리하게 처리하는 기능입니다. 스프라이트 모드1과2가 있습니다.

● **페이지(PAGE)모드**

X-Ⅱ가 가지고 있는 128K 바이트의 비디오 메모리를 여러개의 페이지로 나누어서 사용합니다.
즉, 표시 페이지와 실행페이지로 구분하여 화면에는 표시 페이지의 내용을 나타내면서, 프로그램의 실행결과는 실행페이지에 기록합니다. 이렇게 한 뒤에 표시페이지와 실행페이지를 서로 교환하면 화면의 전환이 매우 빠르게 됩니다. (SET PAGE 명령참조)

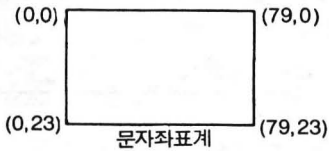
● **인터레이스(INTERLACE, 비월주사)모드**

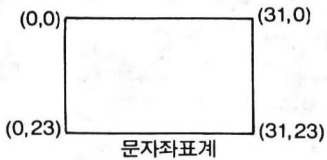
X-Ⅱ의 화면은 보통 세로 212개의 주사선을 이용하고 있으나, 인터레이스 모드를 사용하여 424개의 주사선을 이용함으로써, 세로방향의 해상도를 2배로 증가시킬 수 있습니다. 즉 처음 수직기간(1/30초)에는 짝수번째의 주사선에 화면의 내용을 표시하고, 다음 수직기간에는 홀수번째의 주사선에 다른 내용을 표시합니다. 이렇게 212개의 수평라인을 두번에 걸쳐 표시합니다.
(SCREEN 명령참조)
그러나, 이 모드를 사용할 때는 인터레이스 모드가 가능한 모니터(잔상시간이 긴 모니터)를 사용해야 합니다.

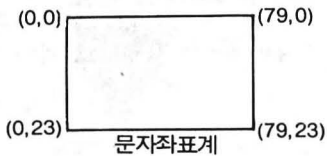
● **화면복사(COPY)**

화면의 내용은 비디오 메모리에 기록되어 있습니다. 이 비디오 메모리의 내용을 비디오 메모리의 다른 페이지로 옮길 수 있으며, 또 배열변수를 사용하여 시스템 메모리로 옮기거나 화일을 써서 디스크 드라이브에 옮길 수도 있습니다. 또 그 반대의 기능도 가능합니다.

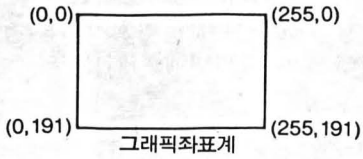
텍스트 화면

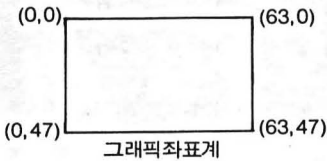
● SCREEN 0 (영문 80자×24줄)

주로 영문자를 표시하는데 사용합니다.
한 문자의 크기는 6×8 화소로 구성됩니다. 이 모드를 실행하면, 처음에는 40문자 표시 형태로 됩니다.
80문자를 나타내려면 다시 WIDTH 80 명령을 실행합니다.

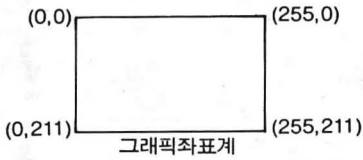
● SCREEN 1 (영문 32자×24줄)

영문자 및 한글을 표시할 수 있읍니다. 영문자는 8×8 화소로 구성되며, 한글은 최대 16×16 화소를 차지합니다.
(즉, 한글은 2줄에 걸쳐서 표시됩니다.)
스프라이트 기능을 사용할 수 있으며, 모드 1이 선택됩니다.

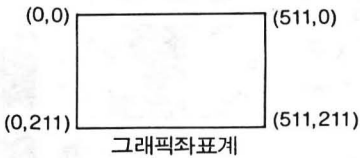
● SCREEN 9 (한글 40자×24줄, 한자 32자×24줄)

영문자 및 한글, 한자를 표시하는데 사용합니다. 영문자는 6×16, 한글은 12×16화소, 한자는 16×16화소로 구성됩니다.
이 모드가 실행되면 한글표시 형태로 되므로, 한자를 함께 표시하려면 다시 WIDTH 64 명령을 실행합니다.
이 모드는 인터레이스 모드이므로, 인터레이스가 가능한 모니터를 사용해야 합니다.

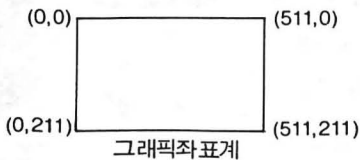
그래픽 화면

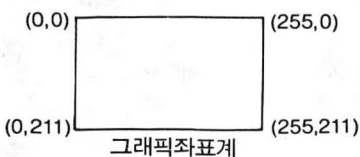
● SCREEN 2, 4 (고해상도 256×192화소)

가로 8 개의 점을 단위로 하여 512색 중에서 16 색을 칠합니다.
스프라이트 기능을 사용할 수 있으며, SCREEN 2에서는 모드 1이, SCREEN 4에서는 모드 2가 선택됩니다.

● SCREEN 3 (저해상도 64×48블록)

4×4 화소를 블록으로 하여, 512색 중에서 16색을 선택하여 색을 칠합니다.
스프라이트 기능은 모드 1이 사용됩니다.

● SCREEN 5 (비트맵 256×212화소)

각 점마다 512색 중에서 16색을 선택하여 색을 칠합니다.
인터레이스 모드가 사용되면, 256×424 화소로 됩니다.
스프라이트 기능은 모드2 가 사용됩니다.
비디오 메모리는 4 (번호 0~3) 페이지로 나누어 사용됩니다.
화면복사 기능이 가능합니다.

● SCREEN 6 (비트맵 512×212화소)

각 점마다 512색 중에서 4 색을 선택하여 색을 칠합니다.
인터레이스 모드가 사용되면, 512×424 화소로 됩니다.
스프라이트 기능은 모드 2가 사용됩니다.
비디오 메모리는 4 (번호 0~3) 페이지로 나누어 사용됩니다.
화면 복사기능이 가능합니다.

● SCREEN 7 (비트맵 512×212 화소)

각 점마다 512색 중에서 16색을 선택하여 색을 칠합니다.
인터레이스 모드가 사용되면, 512×424 화소로 됩니다.
스프라이트 기능은 모드 2가 사용됩니다.
비디오 메모리는 2 (번호 0~1) 페이지로 나누어 사용됩니다.
화면 복사기능이 가능합니다.

● SCREEN 8 (비트맵 256×212화소)

각 점마다 256색을 칠합니다.
인터레이스 모드가 사용되면, 256×424 화소로 됩니다.
스프라이트 기능은 모드 2가 사용됩니다.
비디오 메모리는 2 (번호 0~1) 페이지로 나누어 사용됩니다.
화면 복사기능이 가능합니다.

2. 색상(COLOR)

우리가 그림을 그릴 때는 빠레트에서 물감을 혼합하여, 여러가지 색을 만들어 사용합니다.
X-II 에서는 이와 같은 빠레트가 16개 준비되어 있읍니다. 각각의 빠레트에는 빛의 삼원색(적, 녹, 청색)을 8 단계의 밝기로써 서로 혼합하여 512(8×8×8)가지의 색상을 만들 수 있읍니다.
베이직 명령에서 사용하고 있는 색상 번호는 고정된 색상이 아니라 실제는 칼라 빠레트 번호이므로, 각각의 칼라 빠레트의 색을 변경함으로써, 512색 중에서 16가지의 색상을 동시에 화면에 나타낼 수 있읍니다.
예를들어 X-II 베이직에서는 시스템 초기상태 또는COLOR=NEW 명령으로 아래와 같이 각 빠레트에 적, 녹, 청색을 혼합하여 초기상태의 색상으로 합니다.

빠레트 번호	색 상	적색(R)의 밝기	녹색(G)의 밝기	청색(B)의밝기
0	투명	0	0	0
1	검정색	0	0	0
2	녹색	1	6	1
3	밝은 녹색	3	7	3
4	진한 청색	1	1	7
5	밝은 청색	2	3	7
6	진한 적색	5	1	1
7	하늘색	2	6	7
8	적색	7	1	1
9	밝은 적색	7	3	3
10	진한 황색	6	6	1
11	밝은 황색	6	6	4
12	진한 녹색	1	4	1
13	보라색	6	2	5
14	회색	5	5	5
15	흰색	7	7	7

그러나, 사용되는 칼라 빠레트의 수효는 각 화면 모드에 따라 다릅니다.
SCREEN 0, 1, 2, 3, 4, 5, 7은 16개의 빠레트를 모두 사용하지만, SCREEN 6은 4개의 빠레트만을 사용합니다.
각각의 빠레트는 COLOR= 명령을 사용하여 색상을 변경시킬 수 있읍니다. (명령어편의 COLOR= 명령참조)

예) 10 SCREEN 5
　20 COLOR=(3, 7, 2, 3) ; 색 번호(빠레트 번호) 3에 적색의 밝기를 7, 녹색을 2, 청색을 3으로 하면, 색번호 3은 주황색으로 됩니다.
　30 COLOR=(15, 1, 1, 1) ; 색 번호 15에 적색의 밝기를 1, 녹색을 1, 청색을 1로 하면, 색 번호 15는 검정색으로 됩니다.
　40 COLOR=(12, 0, 0, 7) ; 색 번호 12에 적색의 밝기를 0, 녹색을 0, 청색을 7로 하면, 색 번호 12는 파란색으로 됩니다.
　50 COLOR 15, 3, 12 : CLS ; 화면의 표시색을 검정색, 바탕색을 주황색, 주변색을 파란색으로 나타냅니다.
　60 LINE(10, 10)—(30, 30), , BF
　70 GOTO 70

한편, SCREEN 8은 칼라 빠레트를 사용하지 않고, 직접 녹, 적, 청의 밝기를 혼합하여 256가지의 색상을 동시에 화면에 나타낼 수 있는 강력한 칼라 그래픽 기능을 가지고 있읍니다.
적, 녹색의 밝기는 0~7의 8 단계이고, 청색의 밝기는 4~7의 4 단계입니다.
색상 번호와 색상의 관계는 다음과 같읍니다.

색상번호 (10진수)	2진수		녹색의 밝기	적색의 밝기	청색의 밝기
	$b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$		$b_7\ b_6\ b_5$	$b_4\ b_3\ b_2$	$b_1\ b_0$
227	→1 1 1 0 0 0 1 1 →		1 1 1 ,	0 0 0 ,	1 1 1 → 하늘색 ↓ 항상 1로 설정됩니다.

예)
```
10 SCREEN 8
20 COLOR 248, 28, 3 : CLS
30 LINE(10, 10)-(30, 30), , BF
40 GOTO 40
```
; 248(2진수 11111000)은 노란색을, 28(2진수 00011100)은 주황색을, 3(2진수 00000011)은 파란색을 나타냅니다.
따라서 표시색은 노란색, 바탕색은 주황색, 주변색은 파란색입니다.

녹, 적, 청색의 밝기가 결정되면, 색상 번호는 다음과 같은 식으로 구할 수 있읍니다.
색상번호＝32×(녹색의 밝기)＋4×(적색의 밝기)＋(청색의 밝기)

예)
```
10 SCREEN 8
20 A=32*4+4*7+7 : COLOR A, 28, 3 : CLS ; A에는 163이 대입됩니다. 즉 하늘색을 지정하고,
                                          이것을 표시색으로 사용합니다.
30 LINE(10, 10)-(30, 30) , , BF
40 GOTO 40
```

2 해설편

● **색상의 논리연산**

X-II 에서는 비트맵 그래픽(SCREEN 5~8) 모드에서, 그래픽 명령 또는 화면 복사에 관한 명령 중에서 지정하는 색상과 현재 화면에 표시되는 색상의 논리 연산을 하여, 그 결과를 색상으로 취할 수 있읍니다. 즉 각각의 색번호를 2진수로 하여 각 비트에 대한 논리 연산 결과를 얻읍니다.
색상의 논리 연산에 사용되는 기호는 다음과 같읍니다.

논리연산 기 호	내 용	예			
		지정색	현재색	논리연산	색 상
PSET	지정색	6	4	0110 PSET =0110	6 (진한적색)
PRESET	NOT (지정색)	6	4	0110 PRESET =1001	9 (밝은적색)
AND	(지정색) AND (현재색)	6	4	0110 AND 0100=0100	4 (진한청색)
OR	(지정색) OR (현재색)	6	4	0110 OR 0100=0110	6 (진한적색)
XOR	(지정색) XOR (현재색)	6	4	0110 XOR 0100=0010	2 (녹 색)
TPSET	지정색이 0이면 현재색	0	4	TPSET 0100=0100	4 (진한청색)
	아니면 지정색	6	4	0110 TPSET =0110	6 (진한적색)
TPRESET	지정색이 0이면 현재색	0	4	TPRESET 0100=0100	4 (진한청색)
	아니면 NOT (지정색)	6	4	0110 TPRESET =1001	9 (밝은적색)
TAND	지정색이 0이면 현재색	0	4	TAND 0100=0100	4 (진한청색)
	아니면(지정색) AND (현재색)	6	4	0110 TAND 0100=0100	4 (진한청색)
TOR	지정색이 0이면 현재색	0	4	TOR 0100=0100	4 (진한청색)
	아니면(지정색) OR (현재색)	6	4	0110 TOR 0100=0110	6 (진한적색)
TXOR	지정색이 0이면 현재색	0	4	TXOR 0100=0100	4 (진한청색)
	아니면(지정색) XOR (현재색)	6	4	0110 TXOR 0100=0010	2 (녹 색)

논리연산 기호를 생략하면, 화면에는 지정한 색이 표시됩니다.
각 논리연산 기호 앞에 「T」가 붙으면, 지정색이 투명(색상번호 0) 일 경우에는 현재의 색을 그대로 화면에 표시하고, 그렇지 않으면 현재의 색과 지정색을 논리연산하여 그 결과를 화면에 표시합니다.

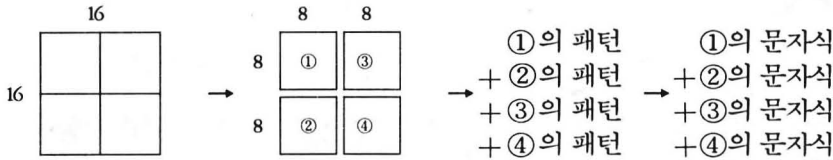
예) 10 COLOR 15, 4, 7 : SCREEN 5
20 LINE(70, 60) – (170, 120), 8, BF, AND ; 현재의 색 (바탕색) 4와 지정한 색 8을 AND하면 투명이 표시됩니다.
30 LINE(80, 70) – (160, 110), 8, BF, OR ; 현재의 색 0 (투명) 과 8을 OR하면 빨간색(8)이 표시됩니다.
40 LINE(70, 60) – (80, 80), 0, BF, TOR ; 지정한 색이 투명이므로, 사각형은 현재의 색으로 표시됩니다. 따라서 사각형은 화면에 나타나지 않읍니다.
50 LINE(90, 80) – (150, 100), 2, BF, TOR ; 지정한 색 2를 현재의 색과 논리연산을 한 결과 (10 : 노란색) 가 표시됩니다.
60 GOTO 60

3. 스프라이트(SPRITE)

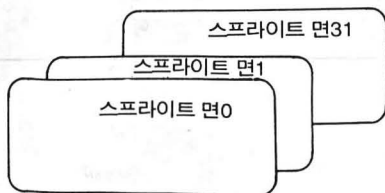
스프라이트는 화면에 작은 그림을 만들어서, 이것을 아주 손쉽게 움직일 수 있는 기능입니다. 이제, 그림의 모양(패턴)을 만들어서 화면에 나타내는 방법을 설명합니다.

패턴을 만드는 방법

패턴의 기본 구성은 8×8 화소입니다. 수평으로 8화소를 1문자로 하여, 패턴마다 8문자로 구성됩니다. 이 때, 형상이 표시되는 화소는 1로, 표시되지 않을 화소는 0으로 나타냅니다.
아래 그림과 같은 스프라이트 패턴의 대응식을 여러가지 방법으로 나타냅니다.

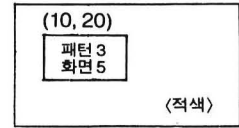
16×16 화소의 패턴은 앞에서 설명한 8×8 화소의 패턴을 아래 그림과 같은 순서로 배열합니다.
그러므로 이 패턴을 만들려면 최대 32문자가 필요합니다.

SPRITE$ (패턴 번호)

위와 같이 만든 패턴에 대한 문자식은 SPRITE$으로 임의의 패턴 번호를 지정하여 정의하여 놓습니다.
이 때, 지정할 수 있는 패턴 번호는
8×8 화소일 때 SPRITE$(0)~SPRITE$(255)
16×16 화소일 때 SPRITE$(0)~SPRITE$(63)의 범위입니다.

스프라이트의 표시

화면에는 0~31의 번호가 붙은 32개의 스프라이트 면을 표시할 수 있으며, 면마다 하나의 스프라이트 패턴을 나타낼 수 있습니다.
스프라이트끼리 겹쳐지면, 면번호가 작은 스프라이트가 위쪽에 있는 것처럼 보입니다.

정의된 패턴을 실제로 화면에 나타내는 명령이 PUT SPRITE입니다.

PUT SPRITE 면번호, (좌표), 색번호, 패턴번호

이 명령이 실행되면 면 번호가 할당된 스프라이트가 화면의 (좌표) 로 지정된 곳에 나타납니다. 이 스프라이트에는 패턴번호로 지정된 패턴의 형상이 색번호에 의한 색상으로 나타나게 됩니다.
예를 들어,
PUT SPRITE 5, (10, 20), 8, 3이면 정의된 패턴 번호 3의 스프라이트가 면 번호를 5로 하여 화면 좌표 (10, 20)에 나타납니다. 색번호는 8이므로 패턴의 색상은 적색으로 됩니다.

스프라이트 패턴의 크기는 8×8 또는 16×16으로 정의되나, 화면에 표시할 때에는 크기를 확대할 수 있읍니다. 스프라이트 확대모드로 되면, 패턴의 각 점은 가로 및 세로로 2개의 점, 즉 실제로 4개의 점에 표시됩니다.

SCREEN 모드, 스프라이트 크기

위 SCREEN 명령에서 「스프라이트 크기」는 0~3의 범위이며, 각각의 다음과 같은 의미를 가집니다.
0 : 8×8 패턴
1 : 8 ×8 패턴의 확대
2 : 16×16 패턴
3 : 16×16 패턴의 확대

스프라이트의 색상

스프라이트 패턴의 색상은 색번호로 지정하며, 이것은 각 화면 모드의 색번호와 같습니다. 즉 스프라이트의 표시색은 칼라 빠레트의 값으로 결정됩니다. 그러나 SCREEN 8에서의 스프라이트 패턴의 표시색은 다음과 같은 고정된 색입니다.

칼라번호	녹색(G)의 밝기	빨간색(R)의 밝기	파란색(B)의 밝기	색 상
0	0	0	0	검정색
1	0	0	2	진한 청색
2	0	3	0	진한 적색
3	0	3	2	진한 보라색
4	3	0	0	진한 녹색
5	3	0	2	진한 하늘색
6	3	3	0	진한 노란색
7	3	3	2	회색
8	4	7	2	살색
9	0	0	7	청색
10	0	7	0	적색
11	0	7	7	보라색
12	7	0	0	녹색
13	7	0	7	하늘색
14	7	7	0	노란색
15	7	7	7	흰색

스프라이트 모드

스프라이트는 SCREEN 0과 9를 제외한 모드에서 사용할 수 있으며, X-II에서는 두 가지의 스프라이트 표시 모드가 있습니다.

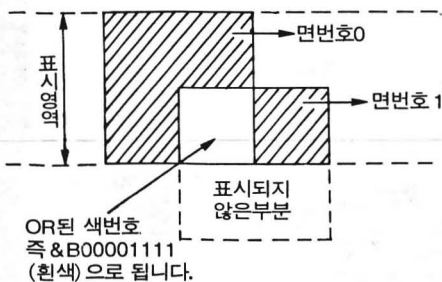
● **스프라이트 모드 1 (SCREEN 1～3)**
스프라이트는 우선도(스프라이트 면 번호가 작은것)가 높은 순으로 화면의 한 수평선상에 4개까지 표시되며, 패턴은 지정된 색상으로 칠해집니다.

● **스프라이트 모드 2 (SCREEN 4～8)**
스프라이트는 우선도가 높은 순으로 화면의 한 수평선상에 8개까지 표시되며, 각 패턴은 가로의 점들을 단위로 하여, 세로로 한 줄씩 다른 색을 칠할 수도 있고, 각 스프라이트의 이동이나 스프라이트끼리 겹치는 부분에 대한 색상의 논리 연산을 행할 수도 있습니다.

즉 다음과 같은 명령을 이용하면,

COLOR SPRITE $ (〈면번호〉) = 〈식〉

〈식〉은 문자의 조합으로 각 문자는 순서대로 스프라이트 패턴의 가로줄에 대응됩니다.
각 문자를 2진수로 나타내면, 각 비트는 아래와 같은 의미를 가지고 있습니다.

b_7 : 1인 경우, 스프라이트가 32화소 좌측으로 이동되어 표시됩니다.
b_6 : 1인 경우, 스프라이트의 우선 순위와 충돌을 무시하며, 스프라이트가 겹쳐졌을 경우는 색번호의 OR 한 값을 화면에 표시합니다. 또 지정된 스프라이트는 연속되는 하나 작은 번호와 함께 이동하며, 화면의 표시 영역은 같은 수평라인 내에서만 표시됩니다.
b_5 : 1인 경우, 스프라이트의 충돌을 무시합니다.
b_4 : 사용하지 않습니다.
b_3～b_0 : 칼라 빠레트 번호

예를 들어, b_6이 1인 경우를 살펴보면,
면번호 0이 &B00000111, 면번호 1이 &B01001000인 때, 두개의 스프라이트가 그림과 같이 겹쳤을 경우를 나타낸 것입니다. 이때, 면번호 0의 스프라이트를 이동시키면, 면번호 1인 스프라이트도 함께 이동됩니다.

	b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0	
면번호0	:0 0 0 0 0 1 1 1	하늘색
면번호1	:0 1 0 0 1 0 0 0	적색
겹치는 부분의 색상:	1 1 1 1	흰색

4. 수퍼임포즈, 디지타이즈

● 수퍼임포즈란 컴퓨터의 영상과 외부 영상신호(TV나 VTR등의 영상신호) 를 합성하여 표시하는 기능입니다. 또한 컴퓨터의 음성과 외부의 음성 신호를 합성할 수 있읍니다.
이 기능을 이용하여 비디오 편집이나 자막처리 등에 활용할 수 있읍니다.

● 디지타이즈란 컴퓨터가 외부영상신호(아날로그 신호)를 받아들여 컴퓨터가 처리할 수 있는 정보(디지탈 신호)로 바꾸어 주는 기능입니다.
이 기능을 이용하여 외부영상신호를 읽어 디스켓에 그림 정보를 저장할 수도 있고, 또한 이 그림을 확대, 축소, 카피할 수도 있읍니다.

● 수퍼임포즈 기능과 디지타이즈 기능은 베이직 명령문「SET VIDEO」나「COPY SCREEN」문 등으로 설정할 수 있읍니다.
컴퓨터 화면과 외부 비디오 화면, 그리고 수퍼임포즈 된 화면으로의 변환은「SET VIDEO」 명령으로도 가능하지만 [CTRL] 키와 펑션키(F1~F3)를 함께 눌러서 변환할 수도 있읍니다.

	모 드	베이직 명령
[CTRL] + [F1]	컴퓨터	Set Video 0, 0,, 0, 0,, 1
[CTRL] + [F2]	수퍼임포즈	Set Video 2, 0,, 1, 3,, 1
[CTRL] + [F3]	외부 비디오	Set Video 3, 0,, 1, 3,, 1

● 수퍼임포즈를 할 때에는 주변색을 반드시 투명색(색번호"0")으로 해야만 합니다.
이 상태에서 배경색이나 표시색을 투명색으로 지정한 부분만 외부 비디오 신호가 나타나고 다른 색으로 지정한 부분에는 컴퓨터의 화면이 표시됩니다.
「COPY SCREEN」 명령을 사용하여 외부 영상신호를 디지타이즈하여 비디오 메모리(VRAM)로 읽어들일 경우, 이 명령을 실행하기 전에「VDP(7)=255」라는 명령으로 VDP의 칼라 레지스터를 초기 설정해 주십시오.

다음은 외부 비디오 신호를 디지타이즈하여 정지화면으로 만드는 프로그램입니다.

```
10 SCREEN 8
20 SET VIDEO 3, 0, 0, 1, 3, 1, 1
30 VDP(7)=255
40 IF INKEY$="" THEN 40
50 COPY SCREEN 0
60 SET VIDEO 1
70 IF INKEY$="" THEN 70
80 GOTO 20
```

또, 칼라버스를 입력모드로 지정하면 연속적으로 디지타이즈 할 수 있읍니다.

```
10 SCREEN 8
20 ON STOP GOSUB 110
30 STOP ON
40 SET VIDEO 1, 0, 1, 1, 3, 1, 1
50 VDP(7)=255
60 IF INKEY$=" "THEN 60
70 COPY SCREEN 0
80 SET VIDEO ,, 0
90 IF INKEY$="" THEN 90
100 GOTO 40
110 SET VIDEO,, 0
120 END
```


주의 수퍼임포즈, 디지타이즈 기능은 X-Ⅱ의 본체 내부에 옵션보드로 내장되어 있읍니다.
따라서 이 옵션보드가 내장되지 않은 시스템에서는 수퍼임포즈, 디지타이즈 기능이 동작되지 않습니다.
만약 수퍼임포즈 옵션보드를 시스템에 장착하고 싶은 경우에는 구입하신 전문점이나 가까운 서어비스 센터에 문의하여 주십시오.

5. 시계기능

X-II에는 시계 기능이 내장되어 있읍니다. 시계기능은 건전지로 백업(Back-Up)되어, 전원을 꺼도 동작되므로 현재의 시각과 날짜를 알 수 있읍니다. 만약, 시각이나 날짜가 틀리면, 건전지의 수명이 다한 것이므로, 가까운 서어비스센터에 문의하여, 새 건전지로 바꾸어 주십시오.

● 시스템에 시각과 날짜를 기억시키는 명령은 SET TIME, SET DATE입니다.
(명령어편의 SET TIME, SET DATE 명령 참조)

예) SET TIME "12 : 05 : 40" ; 시스템에 12시 5분 40초를 기억시킵니다.
SET DATE "87/03/19" ; 시스템에 87년 3월 19일을 기억시킵니다.

● 시스템에서 시각과 날짜를 읽어내는 명령은 GET TIME, GET DATE입니다.
(명령어편의 GET TIME, GET DATE명령 참조)

예) GET TIME X$: PRINT X$; 시스템에 기억되어 있는 시각을 읽어내어 X$에 대입하고, 화면에 표시합니다.
GET DATE A$: PRINT A$; 시스템에 기억되어 있는 날짜를 읽어내어 A$에 대입하고, 화면에 표시합니다.

● 알람(ALARM)기능
위 명령에 ,A를 지정하면, 알람 기능으로 됩니다.
시각과 날짜를 동시에 알람으로 기억시킬 경우는 반드시 시각을 먼저 기억시킨 후 날짜를 기억시켜야 합니다.
시각이 알람으로 기억될 경우, 분 단위로 기억되기 때문에 초는 무시됩니다. 또 날짜가 알람으로 기억될 경우, 년도와 월은 기억되지 않읍니다.

주의 알람기능은 시스템에 따라 동작되지 않을 수도 있읍니다.

6. 시스템 상태 설정기능

X-Ⅱ는 시계 기능에 부가된 메모리에 시스템의 초기 상태를 설정할 수 있습니다. 이 메모리의 내용은 전원이 꺼져도 건전지에 의해 백업(Back-Up)되어, 그 데이타가 계속 유지되므로, 사용자가 편리하도록 다음과 같이 시스템에 초기 상태를 선택할 수 있습니다.

화면의 위치변경

모니터에 표시되는 화면(바탕면)의 위치를 점 단위로 상하좌우로 변경시킬 수 있습니다.
컴퓨터가 화면에 표시하는 내용이 모니터에 의해 가려지는 경우에 모니터의 화면을 조금 이동시켜, 이것을 피할 수 있습니다.

○**베이직 명령** : SET ADJUST

비프(BEEP)음 설정

컴퓨터가 내는 비프음의 음색과 음량을 선택할 수 있습니다.
음색과 음량은 각각 4종류가 있으며, 그 중 한 가지를 설정할 수 있습니다.

○**베이직 명령** : SET BEEP

타이틀(TITLE) 설정

시스템 초기 상태에 화면의 타이틀을 지정할 수 있습니다.
타이틀은 6문자까지 나타낼 수 있으며 타이틀이 6문자이면, 임의의 키를 눌러야만 다음 화면으로 넘어 갑니다.

○**베이직 명령** : SET TITLE

암호(PASSWORD) 설정

시스템을 동작시키는 암호를 설정할 수 있습니다. 암호는 255문자까지이나, 너무 많이 지정하면 기억하기 힘드므로 적당하게 설정하십시오.
시스템 초기 상태에서, 이전에 암호가 설정되어 있으면, 화면「PASSWORD」가 표시되면서 암호 입력 대기상태가 됩니다. 이 때, 올바른 암호를 입력하고 [RETURN] 을 누르면, 시스템이 다음 동작을 하게 됩니다. 만약 암호가 틀렸으면 몇 번이고 다시 입력시키십시오.
그러나 암호가 전혀 기억나지 않을 때는 [GRAPH] 와 [STOP] 키를 함께 누르면서 전원을 켜면, 암호의 입력을 거치지 않고 바로 통과할 수 있습니다.

○**베이직 명령** : SET PASSWORD

프롬프트(PROMPT)의 변경

시스템의 명령 대기 상태를 알려주는 프롬프트 메세지「Ok」를 6문자 이내에서 다른 문자로 변경할 수 있습니다.

○**베이직 명령** : SET PROMPT

> [주의] PASSWORD, PROMPT, TITLE을 동시에 함께 지정할 수 없으며, 만약 함께 지정해 주면, 맨 마지막에 지정된 기능만이 유효하게 됩니다.

화면모드의 설정

시스템 초기 상태의 명령 대기상태 화면은 SCREEN 0로 설정되어 있습니다. 이것을 텍스트 모드 중의 화면 모드로 변경할 수 있습니다. 화면 모드와 함께 변경시킬 수 있는 파라메터는 다음과 같습니다.

- 화면모드
- 한 행의 문자 수
- 색상
- 기능키 스위치
- 키 클릭 스위치
- 데이타 레코더의 전송 속도
- 인터레이스 모드

○**베이직 명령** : SET SCREEN

7. 화일(FILE)

화일이란, 의미를 가진 정보의 집합입니다.
메모리 디스크나 외부기억장치에 저장된 프로그램이나 데이타는 화일로 취급됩니다.

화일번호

프로그램 중에서 화일을 취급할 때는, 먼저 OPEN(오픈) 명령으로 화일의 사용개시를 선언하고, 화일에 대응하는 번호를 붙여 주어, 번호와 대응하는 입출력 영역을 확보해야 합니다. 이 번호를 화일번호라고 부릅니다.
이 후에 입출력되는 화일은 화일 번호에 의해서 지정된 화일로 입출력됩니다.
(명령어편의 OPEN, PRINT #, INPUT # 등을 참조하십시오.)

화일을 취급하는 장소

화일의 입출력을 행하는 장소로는 다음과 같은 곳이 있읍니다.

화일을 사용하는 장소	장 치 명	기 능	비 고
데이타 레코더	"CAS:"	입출력장치	외부기억장치
플로피 디스크 드라이브(FDD)	"A:" 또는 "B:" 등	입출력장치	외부기억장치
퀵 디스크 드라이브(QD)	"QD:" 또는 "QD 0:"	입출력장치	외부기억장치
메모리 디스크	"MEM:" 또는 "MEM 0:"	입출력장치	
텍스트 화면	"CRT:"	출력장치	
그래픽 화면	"GRP:"	출력장치	
프린터	"LPT:"	출력장치	

외부기억장치의 장치명은 생략할 수도 있읍니다.
데이타 레코더와 디스크 드라이브가 함께 연결된 때, 장치명을 생략하면, 디스크 드라이브 중 현재 사용중인 드라이브가 선택됩니다.

화일명

화일명은 화일에 붙여주는 이름으로, 데이타 레코더나 FDD, QD 등의 외부기억 장치에 입출력을 행할 때 필요하며, 그 밖의 경우에는 생략할 수도 있읍니다. 화일명은 영숫자나 한글로 구성되며, 첫 글자는 반드시 문자이어야 합니다.
또 쉼표(,), 마침표(.), 문자코드0 부터 &H1F까지의 문자는 화일명에 포함시킬 수 없읍니다.

- 데이타 레코더에서는 화일명을 6문자까지 사용할 수 있으며, 그 이상의 문자는 무시됩니다.
- FDD나 QD의 화일명에는 타잎명을 붙일 수 있읍니다. 화일명과 타잎명은 · (마침표) 로 구분되며, 타잎명은 3문자 이내이어야 합니다.

8. 데이타 레코더

컴퓨터에 입력시킨 프로그램은 본체의 전원을 끄거나 RESET 을 누르면 지워집니다. 프로그램을 자주 사용하거나 계속 확장 또는 수정하여야 할 필요가 있을 때 프로그램을 저장해 두는 역할을 하는 외부기억장치가 있습니다.
외부기억장치로는 데이타 레코더, 플로피 디스크 드라이브(FDD), 퀵 디스크 드라이브(QD)가 있읍니다.
프로그램을 보존하거나 읽어들이는 속도가 빨라 사용하기 편리한 플로피 디스크 드라이브와 퀵 디스크 드라이브에 대한 사용법은 각 설명서를 참조해 주십시오.
여기서는 데이타 레코더에 프로그램을 보존하거나 읽어들이는 방법에 대해서 설명하겠읍니다.

프로그램의 보존 (SAVE)

우선 데이타 레코더(DPR-600)를 본체에 연결합니다.
프로그램을 테이프에 보존하기 위해서는 CSAVE 명령을 사용하면 됩니다. 이때 테이프의 위치를 확인합니다. 즉 카운터의 위치를 알아둘 필요가 있읍니다.
다음, 데이타 레코더를 녹음상태로 해놓고

CSAVE "SAMPLE" RETURN

을 치면, 테이프가 돌기 시작하면서 프로그램은 테이프에 「SAMPLE」이라는 화일명으로 기록됩니다.
테이프에 기록이 끝나면, 테이프의 회전은 중지되고 화면에 「Ok」가 나타납니다.
리모트 단자가 없는 경우 녹음이 끝나도 동작은 멈추지 않읍니다. 따라서 사용자가 데이타 레코더의 STOP BUTTON을 눌러서 정지시켜야 합니다.

프로그램의 확인

프로그램이 테이프에 제대로 보존되어 있는지를 확인하기 위해서는 테이프를 프로그램의 처음 부분으로 되돌린 후

CLOAD ? "SAMPLE" RETURN

을 치면, 데이타 레코더가 「SAMPLE」이라는 프로그램을 찾기 시작합니다. 만일 테이프에 「SAMPLE」 프로그램 앞에 「ABC」라는 프로그램이 기억되어 있다면,

CLOAD ? "SAMPLE"
Skip : ABC

로 되고, 「SAMPLE」 프로그램을 테이프로부터 찾았다면

Found : SAMPLE

이 화면에 표시되고, 이상이 없으면,

Ok

가 표시되면서 테이프가 정지됩니다. 이상이 있다면,

Verify error
Ok

로 표시됩니다. 이때는 다시 테이프에 프로그램을 저장시켜야 합니다.

프로그램을 읽어들임(LOAD)

프로그램을 테이프로부터 컴퓨터에 옮기기 위해서는 CLOAD 명령을 사용하며, 그 순서는 CLOAD ? 명령의 사용 순서와 비슷합니다. 다시 말하면, 다음과 같이 됩니다.

1. 테이프를 프로그램의 처음의 위치에 맞추어 놓읍니다.
2. PLAY BUTTON을 누릅니다.
3. CLOAD 명령을 입력합니다.

화일 「SAMPLE」의 내용을 읽어들일려면, 다음과 같이 입력합니다. 즉

CLOAD "SAMPLE" [RETURN]

화일이 발견되면,

Found : SAMPLE

로 화면에 표시되며, 다 읽어들이면,

OK

가 표시되고, 테이프가 정지합니다.
리모트 단자가 없는 경우에는 위의 1, 3, 2의 순서로 동작시켜야 하며, 다 읽어들인 후에도 테이프는 정지하지 않읍니다. 프로그램이 잘 저장되었는지를 보기 위해서 LIST 명령을 사용하여 프로그램의 내용을 확인하십시오.

[주의] CLOAD를 할 경우 화일명을 생략하면, 맨처음 찾아지는 프로그램이 읽혀집니다.

FF와 REWIND

리모트 상태에서는 FF나 REWIND가 되지 않읍니다. 이때는 MOTOR 명령을 사용함으로써 FF나 REWIND를 할 수 있읍니다. 우선,

MOTOR ON [RETURN]

을 치면 리모트상태가 해제되고, 이 상태에서는 FF나 REWIND를 함으로써 테이프의 위치를 조정할 수 있읍니다. 그 다음,

MOTOR OFF [RETURN]

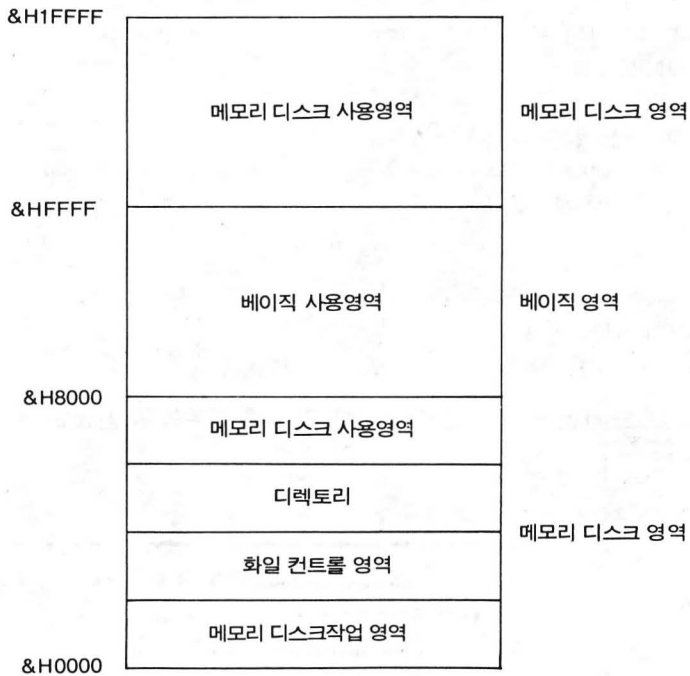
을 입력하면, 리모트 상태가 되고, CLOAD, CSAVE가 가능하게 됩니다.

9. 메모리 디스크(MEMORY-DISK)

MSX 시스템은 보통 64K 바이트의 메모리를 가지고 있으나, 베이직에서는 상반부 32K 바이트만 사용하고 있읍니다. (64K 바이트를 모두 사용하는 경우는 MSX-DOS를 운용할 때입니다.)
메모리 디스크 기능이란 베이직에서 하반부의 32K 바이트의 메모리와 메모리 맵퍼에 있는 64K 바이트의 메모리를 활용하여 마치 외부기억장치처럼 프로그램이나 데이타를 저장하거나 로드할 수 있는 기능입니다.

메모리 디스크의 제원

- 시스템 메모리 영역 : 약 6K 바이트
- 데이타 메모리 영역 : 약 90K 바이트
- 화일 수 : 최대 64개
- 화일 모드 : 시퀀셜 화일(랜덤 화일은 불가능함)
- 장치명 : "MEM:" 또는 "MEM0:"

메모리 디스크의 메모리 맵

메모리 디스크의 사용

● 메모리 디스크 기능을 사용하려면, 먼저 다음과 같이 선언해 주어야 합니다.
CALL MEMINI [RETURN]
이와 같이 하면, 화면에는 "91904 bytes allocated"라고 표시 됩니다.

● 프로그램이나 데이타의 보존 또는 읽어내기
몇 개의 프로그램을 메모리 디스크에 보존해 두고, 필요한 프로그램을 선택해서 실행시킬 수 있습니다. 또 두 개의 프로그램을 합쳐서, 하나의 프로그램으로 할 수도 있습니다.
프로그램을 보존할 때 : SAVE 명령
프로그램을 읽어낼 때 : LOAD 명령
프로그램을 합칠 때 : MERGE 명령

● 데이타의 보존이나 읽어내기
많은 데이타를 취급할 경우, 베이직 사용 영역의 메모리 용량이 부족할 때가 있읍니다. 이와 같은 경우에는 메모리 디스크의 화일에 일시적으로 데이타를 보존하면 보다 많은 데이타를 취급할 수 있읍니다.
메모리 디스크 화일에서의 데이타 입출력은 다음과 같은 순으로 진행됩니다.

OPEN : 메모리 디스크의 화일로부터 데이타를 입출력하기 전에 화일번호를 할당합니다.
PRINT # 또는 INPUT #
: OPEN 명령에서 지정한 화일 번호의 화일에 대하여 데이타의 출력(PRINT #) 또는 입력 (INPUT #)을 실행합니다.
CLOSE : 화일을 닫읍니다. 데이타의 입력이 끝나면 반드시 CLOSE를 해야 합니다.
CLOSE를 하지 않으면 남아있는 데이타가 파괴될 수도 있읍니다.

● 메모리 디스크의 기능을 끝내려면, 다음과 같이 선언해 주어야 합니다.
CALL MEMINI(0) [RETURN]
이와 같이 하면, 화면에는 "No RAM disk"로 표시됩니다.

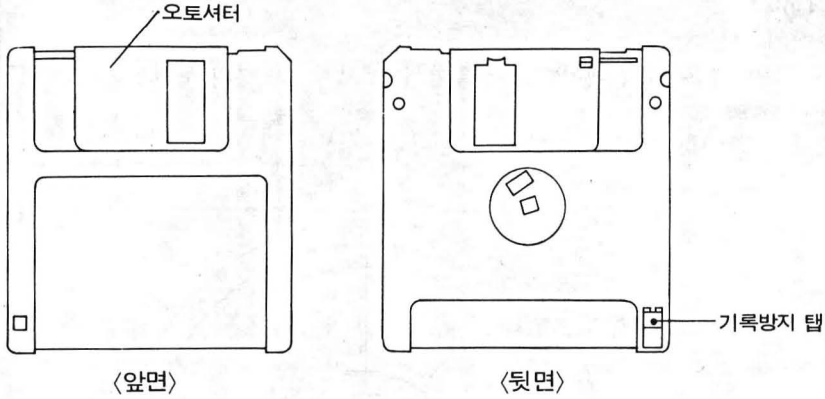
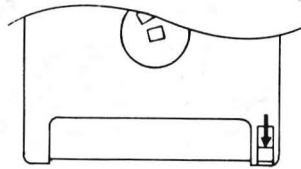
■ **화일명은 디스크 드라이브의 화일명과 같으므로, 화일부분을 참조하여 주십시오.**

메모리 디스크에서 전용으로 사용되는 명령

CALL MEMINI : 메모리 디스크 기능을 선언하고, 메모리 디스크를 초기화 합니다.
CALL MEMINI(0) : 메모리 디스크 기능의 해제를 선언합니다.
CALL MFILES : 메모리 디스크의 화일명을 모두 표시합니다.
CALL MKILL : 메모리 디스크의 화일을 삭제합니다.
CALL MNAME : 메모리 디스크의 화일명을 변경시킵니다.

10. 플로피디스크

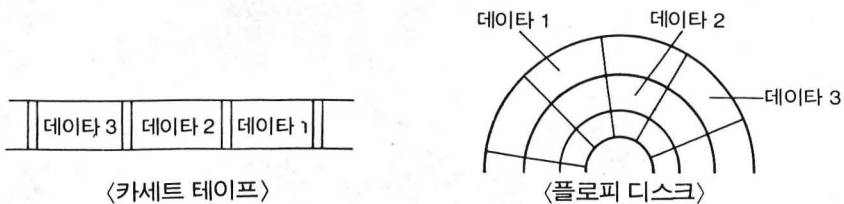
플로피 디스크는 많은 양의 정보를 기억할 수 있는 보조기억 매체로서 디스켓이라고도 합니다.
플로피 디스크는 카세트 테이프보다 프로그램이나 데이타를 빠르고 정확하게 처리할 수 있읍니다.
X-II에서 사용하는 플로피 디스크는 3.5인치 플로피 디스크로 아래 그림과 같은 모양을 하고 있읍니다.

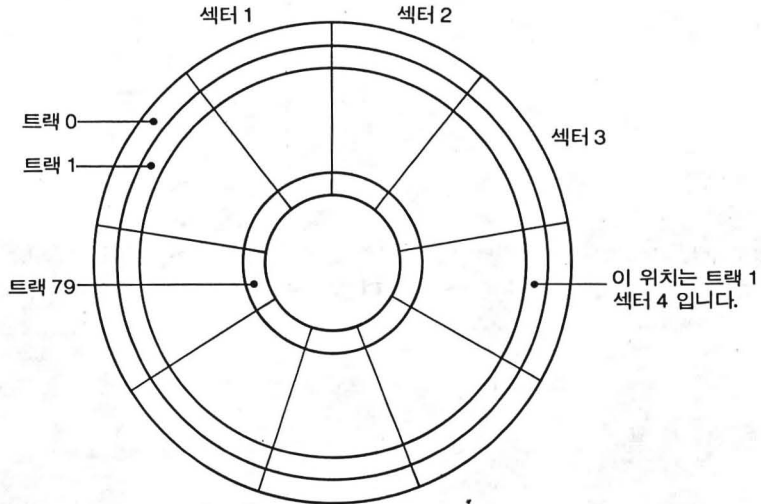
〈앞면〉 〈뒷면〉

플로피 디스크의 뒷면에 있는 기록방지탭은 디스켓에 데이타를 써 넣는 것을 방지할 수 있읍니다.
즉 기록방지 탭을 그림과 같이 아래로 내리면 디스켓에 써 넣는 것이 불가능하고 반대로 위로 올리면 써 넣는 것이 가능합니다.

플로피 디스크를 사용할때는 다음 사항에 주의하여 주십시오.

- 고온, 저온, 직사광선 및 습기가 많은 곳에서 사용하거나, 보관하지 마십시오.
- 디스켓에 커피, 쥬스, 음료수나 화학성물질 등을 엎지르거나, 무거운 물건을 올려 놓지 마십시오.
- 오토 셔터를 손으로 열거나 디스크면을 만지지 마십시오.
- 자기가 있는 곳(스피커나 모터 가까이)에 보관하지 마십시오. 특히 자석은 절대로 가까이 두어서는 안됩니다.
- 디스켓에 데이타를 써 넣거나, 읽어내는 도중에 디스켓을 꺼내거나 전원을 끄지 마십시오. 디스켓에 있는 데이타가 지워지는 수도 있읍니다.

플로피 디스크의 구조

플로피 디스크는 카셋트 테이프처럼 데이타를 차례대로 기록하는 것이 아니고, 몇 개의 영역으로 미리 구분해 놓고 그 영역내에 기록해 갑니다.
이와 같이 기록할 영역을 구분하는 것을 포맽(Format) 이라고 합니다.

〈카세트 테이프〉 〈플로피 디스크〉

아래 그림에서와 같이 포맬된 디스크의 중심으로부터 동심원상의 경계를 트랙이라고 하고 방사상의 경계를 섹터라고 합니다.
양면 배밀도의 경우 디스켓에는 트랙이 양면으로 160트랙, 그리고 섹터는 한트랙 당 9섹터가 있으므로
160×9=1440
즉 1440 섹터가 있읍니다.
또 한 섹터에는 512바이트의 데이타를 기록할 수 있으므로 1장의 포맬된 디스켓에는
1440×512=737280
즉 737280 바이트의 데이타를 기록할 수 있읍니다.

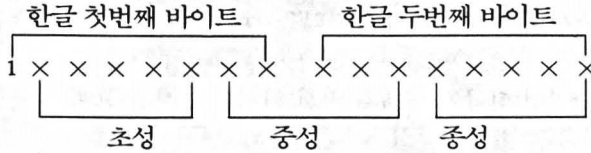
〔그림 1〕

시퀸셜(Sequencial) 화일과 랜덤 엑세스(Ramdom Access)화일

플로피 디스크에 저장할 수 있는 데이타 화일은 크게 나누어 시퀸셜 화일과 랜덤 엑세스 화일로 두 가지 종류가 있읍니다.
시퀸셜 화일이란 데이타를 순차적으로만 처리할 수 있는 화일을 말합니다. 즉 디스켓에 저장되어 있는 정보를 읽어내거나, 써 넣을때 사용자가 임의로 사용할 레코드의 위치를 지정할 수 없고 오직 순서대로만 사용할 수 있읍니다. 이러한 방법은 데이타 레코더에서 정보를 처리하는 방법과 같습니다.
반면, 랜덤 엑세스 화일은 미리 레코더의 길이를 정해놓고 각 레코더에 순서에 관계없이 데이타를 읽어내거나 써 넣을 수가 있읍니다.

11. 한글

X-Ⅱ의 한글은「MSX 한글 Version 3.1」을 채용하였으며「분리형 16비트 코드」로 구성되어 있읍니다.

한글 Ver 3.1의 코드

「분리형 16비트 코드」 체계는 최상위 비트를 1로 세트하고, 나머지 15비트를 5비트씩 나누어서 각각 한글의 초성, 중성, 종성에 대응합니다.

한글소프트웨어 호환성

한글 Ver 3.1은 기존의 MSX2 방식의 한글 Ver 3.0에 한자 처리 기능을 추가한 것으로 한글 Ver 3.0과는 호환성이 있읍니다.
그러나, 한글 Ver 3.1과 MSX의 한글 Ver 2.0(N-바이트 한글코드)은 서로 코드 체계가 다르므로, 한글 코드 자체의 호환성은 없읍니다.
X-Ⅱ에서는 기존 MSX 소프트웨어를 운용할 수 있도록 하드웨어 및 소프트웨어적으로 한글 Ver. 2.0을 위한 아래와 같은 스위칭 기능을 가지고 있읍니다.

- 하드웨어 스위칭
  시스템 초기 상태의 화면에서 [ESC]와 [SELECT]키를 함께 누르면, 한글 Ver 2.0 모드로 동작됩니다.
- 소프트웨어 스위칭
  베이직 명령 대기 상태 또는 베이직 프로그램 작성 중에 한글 코드의 모드를 변경하고 싶을 때는 명령어편의 SET HAN 명령을 참조하십시오.

> [주의] 사용될 한글 코드의 모드가 변환될 뿐, 이 전에 이미 메모리에 저장된 한글 코드는 변환되지 않읍니다.

한글 화면

SCREEN 0 및 1에서는 한글이 두 줄에 걸쳐서 표시되나, SCREEN 9에서는 영문자와 한글이 모두 한 줄에 표시되며, 글자체가 훨씬 세련됩니다. 그러나 SCREEN 9 모드를 이용하려면 인터레이스 모드가 가능한 모니터를 사용해야 합니다.
X-Ⅱ에서 한글 프로그램의 화면 편집은 글자 단위로 이루어집니다. 또한 X-Ⅱ에서는 2메가 비트의 문자롬이 내장되어 있어 SCREEN 9에서「Width 64」로 지정하면 16×16의 명조체 한글을 사용할 수 있읍니다. 또한 SCREEN 5 이상의 비트맵 그래픽 모드에서도 명조체 한글을 사용할 수 있읍니다.

2 해설편

프로그램을 작성할 때의 유의사항

○문자열 연산의 경우, 한글 첫번째 바이트와 두번째 바이트가 뒤바뀌거나, 중간에서 갈라지는 일이 없도록 유의하십시오.

○INPUT$, INKEY$ 명령으로 입력된 문자를 화면에 표시하려면, 최초의 미완성 글자(음소)의 위치를 먼저 지정해 주십시오.
그러나 입력된 미완성 문자를 화면에 표시하지 않을 경우에는 다음 명령을 사용하십시오.

*POKE &HFD0B, 1 : 미완성된 한글은 완성될 때까지 표시되지 않읍니다.
POKE &HFD0B, 0 : 미완성된 한글이 현재의 커서 위치에 표시됩니다.
(이것은 시스템 초기 모드입니다.)

●**프로그램 예 1**

```
10 SCREEN 9 : A$="" : B$="" ; 텍스트 모드 선언
20 LOCATE 10, 10            ; 최초의 미완성 글자가 표시될 위치를 정합니다.
30 B$=INKEY$                ; 키보드로부터 한 문자를 받습니다.
                              이 때, 미완성된 한글은 화면에 표시되지만,
                              B$="" 상태가 됩니다.
40 IF B$="" THEN 30
50 A$=A$+B$                 ; A$에는 지금까지 입력된 코드가 저장됩니다.
60 PRINT B$;                ; 키보드로부터 입력한 완성된 글자를 출력합니다.
70 GOTO 30
```

●**프로그램 예 2**

```
10 SCREEN 5 : A$="" : B$="" ; 그래픽 모드 선언
20 OPEN "GRP :"FOR OUTPUT AS #1
30 PRESET(16, 90)
40 B$=INPUT$(1)
50 A$=A$+B$
60 PRINT #1, B$;
70 GOTO 40
```

○텍스트 모드에서는 비디오 메모리의 6A00H부터 7000H까지 한글의 워크 영역(Work Area)으로 사용하고 있으므로, 비디오 메모리에 데이타를 써넣을 때는 주의하십시오.

프린터 출력모드

X-II에서는 한글 코드를 2종류(2바이트 분리형 또는N-바이트 코드) 운용하므로, 각각에 대하여 프린터 출력 모드를 지정할 수 있읍니다.
명령어편의 SET HAN 명령을 참조하십시오.

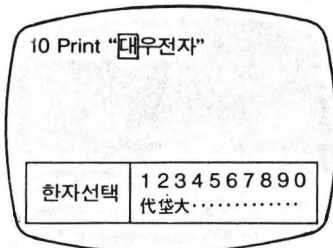
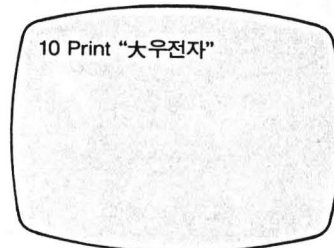
12. 한자

X-Ⅱ에서는 2 메가 비트의 문자롬이 내장되어 있읍니다. 문자롬 안에는 3640자의 고딕체 한자와 2591자의 명조체 한글, 그리고 790자의 영문 및 특수기호가 들어 있읍니다.
X-Ⅱ의 문자롬에 의해서 텍스트 모드와 그래픽 모드에서 한글과 한자를 사용할 수 있으며, 별도의 프로그램에 의해 영문 및 특수기호도 사용할 수 있읍니다.

텍스트 모드

텍스트 모드에서 한자를 사용하기 위해서는 SCREEN 9 모드에서 Width를 64로 지정해 주어야 합니다. Width 64를 지정하면 한글 및 한자를 32×24자로 사용할 수 있읍니다. (Width 64 이하도 사용가능)
한자의 입력은 크게 메뉴입력과 코드입력의 두가지 방식이 있읍니다.

● 한자의 메뉴입력

먼저 SCREEN 9에서 「Width 64」를 설정한 후 한글을 입력합니다. 그 후 한자로 변환하고자 하는 한글의 위치에 커서를 옮긴 후 [SELECT] 키를 치면 화면의 하단에는 현재 커서의 위치에 있는 한글의 음에 해당하는 한자가 표시됩니다.

〈그림 1〉 〈그림 2〉

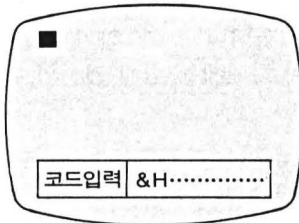
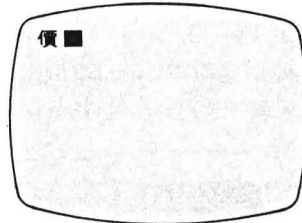
〈그림 1〉은 「대」자에 커서를 옮겨 놓은 후 [SELECT] 키를 누른 경우입니다. 이 상태에서 원하는 한자의 번호(0~9 사이)를 누르면 현재의 커서 위치에 누른 번호에 해당하는 한자가 표시되고 화면 하단의 메뉴화면은 사라지게 됩니다. (그림 2)는 "3"을 누른 경우의 예입니다.

만약 현재의 메뉴 화면에 원하는 한자가 없는 경우, 스페이스 키를 누르면 나머지 한자들이 계속해서 메뉴에 표시되어 집니다.
〈그림 1〉의 상태에서 다음과 같은 키 조작에 의해 한글의 한자 변환을 좀더 편리하고 손쉽게 할 수 있읍니다.

〈그림 1〉의 상태에서 누른 키	기 능
숫자키(0~9)	그 숫자의 하단에 표시된 한자를 현재의 위치에 표시해주고 메뉴화면을 소거한 후 베이직 명령대기 상태로 복귀
스페이스 키	현재의 메뉴화면을 소거하고 아직 표시되지 않은 나머지 한자들을 계속해서 표시
[SHIFT] + 숫자키(0~9)	현재의 커서 위치에 선택된 한자를 표시해주고 커서는 그 다음 한글의 위치에 놓여진 후 다시 그 위치의 한글에 해당하는 한자를 메뉴화면에 표시
[SHIFT] + [SELECT]	현재의 메뉴화면을 소거하고 커서는 그 다음 한글의 위치에 놓여진 후 그 위치의 한글에 해당하는 한자를 메뉴화면에 표시
[SELECT]	현재의 메뉴화면을 소거하고 베이직 명령 대기 상태로 복귀

● 한자의 코드입력

메뉴입력 시에는 먼저 한글을 입력한 후 그 한글의 위치에 커서를 옮겨야하지만 코드 입력 방식은 먼저 한자를 표시할 위치에 커서를 옮기고 난 후 [SHIFT]키와 [SELECT]키를 동시에 누릅니다. 그러면 〈그림 3〉과 같은 화면이 나타나는데, 원하는 한자 코드를 16진수 값으로 입력한 뒤 [RETURN]키를 누르면 현재의 커서 위치에 입력한 코드에 대응하는 한자가 표시되고 메뉴화면이 소거됩니다. (자료편의「한자코드(시스템)」을 참조해 주십시오.)

〈그림 4〉는 한자코드를「D442」로 입력했을 때 입니다. 만약 코드입력 도중 잘못했을 경우에는 [BS]키를 누르면 다시 코드입력을 할 수 있읍니다. 입력한 코드가 한자 코드의 범위를 벗어날 경우에는 화면의 하단에 "한자가 등록되어 있지 않습니다"라는 에러 메시지가 나타납니다.

〈그림 3〉　　〈그림 4〉

> [주의] 한자를 사용할 수 있는 텍스트 모드는 SCREEN 9의 Width 64 (혹은 Width 64 이하) 에서만 가능합니다. 만약 이 모드에서 베이직으로 한자를 입력한 후 SCREEN 0이나 SCREEN 1에서 프로그램의 리스트를 보면 한자를 입력한 부분이 나타나 있지 않게 됩니다. 왜냐하면 이 모드에서는 한자를 표시할 수 없기 때문입니다. 그러나 메모리에는 한자 코드가 기억되어 있습니다. 즉 한자가 삭제된 것이 아니라 단지 표시만 할 수 없는것 뿐입니다.

그래픽 모드

비트맵 그래픽 모드(SCREEN 5~8)에서는 베이직 명령문「PUT HAN」혹은「PRINT #」등으로 한자 및 한글을 표시할 수 있읍니다. (베이직 설명서 편을 참조해 주십시오.)

3

MSX-DOS란 ?

1. MSX-DOS에 대하여

MSX-DOS

MSX-DOS는 MSX 컴퓨터에서 사용하기 위한 디스크 오퍼레이팅 시스템으로서 MS-DOS와 데이타 화일의 호환성이 있고 CP/M과 펑션콜 수준에서 호환성이 있읍니다.

1. MSX-DOS의 개요

MSX-DOS는 64K byte의 메모리(RAM) 용량과 1개 이상의 드라이브를 가지고 있는 MSX 컴퓨터에서 디스크의 화일을 관리하기 위한 오퍼레이팅 시스템입니다.
MSX-DOS는 2개의 화일로서 구성되어 있으며, 내용은 다음과 같습니다.

화 일 명	화일의 기능
MSXDOS.SYS	MSX-DOS 오퍼레이팅 시스템
COMMAND.COM	MSX-DOS 명령어의 처리

표 1-1 MSX-DOS 화일의 구성

2. MSX-DOS의 작동

컴퓨터의 전원을 켜면 MSX-DOS가 메모리 상에 로드되고 시스템은 디스켓에서 COMMAND. COM 화일을 찾아 메모리에 기억시킵니다.
COMMAND. COM 화일은 키보드로부터 입력된 명령어를 해석하여 지정된 프로그램을 실행시키는 프로그램으로서, 명령어 프로세서(Command Processor)라고도 합니다. 명령어 프로세서가 로드되고 난 뒤, 화면에는 다음과 같은 메시지가 나타납니다.

```
MSX-DOS version 1. xx
Copyright 1984 by Microsoft
Command version 1. xx
A>■
```

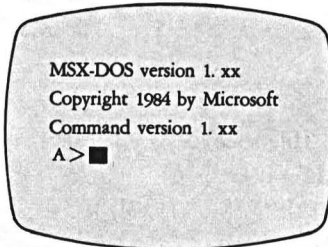



그림 1-1 명령어 프로세서

날짜는 년－월－일의 순서로 입력해야 하며, 숫자 사이의 구분은－(hypen)이나, /(slash)로 합니다.
날짜의 입력이 끝나면 MSX-DOS 프롬프트 "A>■"가 출력되며, "A>" 프롬프트는 MSX-DOS가 명령을 받아들일 수 있는 준비가 되어 있음을 나타냅니다. 여기에서 A는 드라이브 A를 통하여 화일이 처리되는 것을 의미합니다. 이때 B 드라이브를 지정하고자 할 경우에는 "B:"를 입력하면 됩니다.

A>■
A>B : RETURN
B>■ 드라이브 B에서 동작합니다.

주의 디스크 드라이브가 1대인 경우에 A, B 드라이브의 구분은 디스켓을 바꾸어 넣어줌으로서 가능합니다.
이 경우 다음과 같은 메시지가 표시됩니다. 이때 다른 디스켓을 넣고 아무키나 치면 B 드라이브에서 동작됩니다.
Insert diskette for drive B:
and strike a key when ready. ■

MSX-DOS 프롬프트 B＞는 B드라이브를 통하여 화일을 처리한다는 것을 나타냅니다.
화일은 최대 8자를 쓸 수 있는 화일명과 3자 이내로 쓸 수 있는 타잎명으로 지정할 수 있읍니다.
화일명은 대문자 또는 소문자를 구분하지 않고 사용할 수 있으며, 디스켓에는 대문자로 저장됩니다.
MSX-DOS에서는 드라이브명과 함께 화일명의 지정이 가능하며, 지정된 드라이브에서 명령이 수행된 후에는 본래의 드라이브로 되돌아오게 됩니다.
즉 드라이브명이 A로 지정되어 있을 때, B드라이브의 화일에 대하여 명령을 수행시키고자 할 경우 그림 1-2(a)와 같이 드라이브명을 명령어에서 지정할 수 있으며, B드라이브에서 입력된 명령을 수행한 후 A드라이브로 복귀하게 됩니다. 그러나 그림 1-2(b)와 같이 B드라이브를 지정한 후 명령을 수행하는 방법도 있읍니다.

(a) (b)

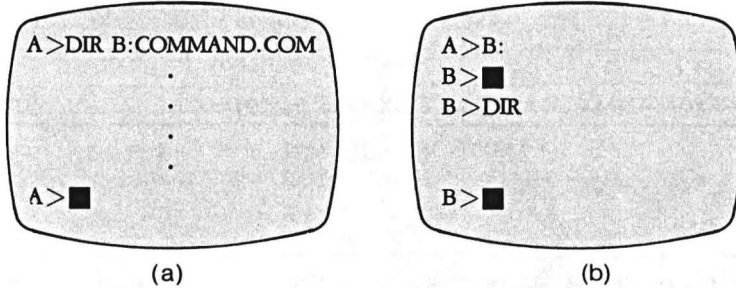



그림 1-2 드라이브명의 지정

화일명과 타잎명은 문자, 숫자 및 그 밖에 다음과 같은 기호등을 사용할 수 있으며, 스페이스(space) 또는 그 밖의 기호는 입력할 수 없읍니다.

A-Z	a-z	0-9	$	#	&
%	,	( )	—	@	
/	∧	{ }	~	!	

〈filespec〉 위에서 설명한 내용은 〈file spec〉으로 정의하며 화일은 다음과 같은 형식을 이용하여 나타낼 수 있읍니다.

〔〈드라이브명 :〉〕〈화일명〉〔〈. 타잎명〉〕

3. 와일드 카드(Wild Card)

MSX-DOS에서는 화일명이나 타잎명의 입력에 사용되는 특수한 문자로서 와일드 카드라고 하는 물음표(?)와 애스터리스크(*)를 사용합니다.

1. "?" 와일드 카드

화일명이나 타잎명에서 "?"는 임의의 문자나 숫자, 기호등을 대신하여 표기될 수 있읍니다.
예를들어, 다음과 같은 MSX-DOS 명령어는 TEST와 RUN 사이의 임의의 문자에 대한 화일명과 타잎명이 COM으로 구성된 화일을 찾아 출력하게 됩니다.

```
DIR TEST?RUN. COM
```

2. "*" 와일드 카드

화일명이나 타잎명에서 "*"는 임의의 문자나 숫자, 기호등의 배열을 대신하여 표기될 수 있읍니다.
예를 들어, 다음과 같은 MSX-DOS 명령어는 TEST 다음에 오는 임의의 문자 배열에 대한 화일명과 COM으로 구성된 타잎명을 찾아 출력하게 됩니다.

```
DIR TEST* .COM
```

화일명과 타잎명에 모두 "*"를 사용하게 되면 디스켓에 기록되어 있는 모든 화일을 처리할 수 있읍니다. 다음의 예는 와일드 카드를 사용한 MSX-DOS 명령어로서 드라이브 A의 디스켓에 있는 모든 화일을 B 드라이버의 디스켓에 복사합니다.

COPY A:*.* B:

드라이브 B의 화일중 타잎명이 COM으로 되어있는 모든 화일을 삭제하고자 할 경우에는 다음과 같이 명령합니다.

DEL B:* .COM

4. 사용불가 화일명

MSX-DOS에서는 다음에 정의하는 화일명 및 타잎명을 특정한 주변장치에 할당하고 있으므로 드라이브에서는 사용할 수 없읍니다.
AUX : 프린터나 드라이브와 같은 주변장치의 입력이나 출력에 관계될 때 사용합니다.
CON : 키보드로부터의 입력이나, 모니터로의 출력에 관계될 때 사용합니다.
LST : 프린터의 출력에 관계될 때 사용합니다.
PRN : LST와 더불어 프린터의 출력에 관계될 때 사용합니다.
NUL : 화일로서 기록되지는 않고, 주어진 화일명과 타잎명으로 입력이나 출력을 하고자할 때 사용합니다.

5. 디렉토리(Directory)

디렉토리는 화일명, 화일의 크기, 디스켓에 있는 화일의 위치, 화일의 생성 날짜 및 변경날짜 등에 관한 정보를 기록하는 영역으로서 "DIR" 이라고 하는 MSX-DOS 명령을 사용하여 디렉토리의 내용을 알 수 있읍니다.

6. MSX-DOS 명령어의 종류

MSX-DOS 명령어에는 다음과 같이 2가지 종류가 있읍니다.
○내부 명령어
○외부 명령어
내부 명령어는 간단하고 자주 쓰이는 명령어로서, 명령어 프로세서의 일부이므로, 명령어 프로그램의 내용을 볼 수는 없으며 명령어 실행만이 가능합니다.
다음은 내부 명령어로서 상세한 내용은 2장에서 설명하였읍니다.

BASIC	DIR	REM	COPY
FORMAT	REN	DATE	MODE
TIME	DEL	PAUSE	TYPE

외부 명령어는 프로그램 화일로서 디스켓에 들어 있으며 xxxxxxxx.COM 또는 xxxxxxxx.BAT의 형태를 가지고 있읍니다. 외부 명령어는 다음과 같은 순서에 의하여 실행됩니다. 그러나 현재 사용중인 디스켓상에 입력된 외부명령어에 대한 화일이 없으면 에러로 처리됩니다.
1. 외부 명령어의 화일명을 입력시킵니다.
2. MSX-DOS는 지정된 드라이브로 부터 화일을 읽어 들입니다.
3. 외부 명령어를 실행합니다.
4. 외부 명령어의 실행이 끝난 후, COMMAND .COM 화일을 다시 읽어 들이고 DOS 프롬프트를 출력합니다.

외부 명령어는 드라이브 사용자가 프로그램할 수 있으며, 화일의 형태는 반드시 COM 혹은 .BAT의 화일 타잎으로 되어야 합니다.

> 참고 외부 명령어를 실행하고자 할 경우 타잎명을 제외한 화일명만을 입력해야 합니다.
> 예) A> LINK.COM : 잘못 사용한 외부 명령어
> 예) A> LINK : 제대로 사용한 외부 명령어

7. 부가명령(Command Option)

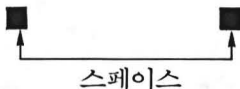
MSX-DOS의 명령어에는 시스템에 필요한 부가명령을 추가하여 사용할 수 있습니다.
부가 명령이 입력되지 않은 명령어인 경우에는 고유의 기능만을 수행하게 됩니다.
다음에 모든 MSX-DOS 명령어의 기본형태를 나타내었읍니다.
부가 명령은 다음과 같은 5개의 형태가 있읍니다.

- **스위치** : MSX-DOS 명령어를 제어하는 부가 명령으로서 "/"로서 구분됩니다.
  예) A >DIR/P
  A >DIR/W
- **화일스펙** : 드라이브명, 화일명, 타잎명으로 구성되며 기본형식은 다음과 같습니다.
  〔〈d : 〉〕〈화일명〉〔〈.EXT〉〕
- **d**: 사용하고자 하는 디스크 드라이브명을 기입합니다.
  예) B >A :
  A >B :
- **화일명** : 디스크 화일의 이름으로서 타잎명을 포함합니다. 화일명은 4 항에서와 같은 주변장치명으로는 입력할 수 없읍니다.
  예) COMMAND.COM
  AUTOEXEC.BAT
- **.EXT** : 화일의 타잎을 지정하는 부분으로서 화일명의 바로 뒤에 위치합니다.

8. MSX-DOS 명령어의 공통사항

다음에 설명하는 12개 사항은 모든 MSX-DOS 명령어에 공통으로 적용됩니다.

1. 명령어는 보통 1개 이상의 부가명령을 가지고 있읍니다.
2. 명령어와 부가명령은 영문 대문자, 소문자 혹은 혼합된 형태로 사용할 수 있읍니다.
3. 명령어와 부가명령은 스페이스, 또는 콤마(,)로서 구분하여 표기 합니다.

   예) COPY COMMAND.COM B : COMMAND.AAA

그 이외에는 세미콜론(;), 등호(=) 혹은 [TAB] 키로서 구분할 수도 있으나, 일반적으로 스페이스를 사용합니다.

4. FILESPEC은 콜론(:)과 피어리드(.)로 구분지어야 합니다.
5. 화일의 수행중 "Strike a key when ready"라는 메시지가 나타나면 [CTRL]+[C]키를 제외한 모든 키를 사용할 수 있읍니다.
6. 화일 처리시 화일의 타잎명만을 입력시킬 수 없으며, 화일명과 함께 사용해야 합니다.
7. [CTRL]+[C]키는 수행중인 명령을 취소할 수 있읍니다.
8. 명령어의 실행은 [RETURN]키를 치면 됩니다.
9. 와일드 카드와 주변장치명은 명령어로 사용할 수 없읍니다.

10. 모니터상에서 많은 양의 내용을 보고자 할 경우, [CTRL]+[S]를 누르면 화면이 일시 정지하며, 다시 한 번 누르면 모니터 출력을 재개하기 시작합니다.
11. MSX-DOS의 에디팅 키(editing key)와 기능 키(function key)는 명령어를 입력할 때 사용할 수 있으며, 상세한 내용은 13항에서 설명하였읍니다.
12. MSX-DOS 프롬프트는 현재 사용중인 드라이브명을 나타냅니다.
    예) A>■

9. 일괄처리 프로그램(Batch processing program)

MSX-DOS에서는 일괄처리 화일(batch file)이라고 하는 특수한 화일이 있읍니다. 이 화일은 여러개의 명령어를 순차적으로 처리하고자 할 때 사용하며 화일명은 임의로 쓸 수 있으나, 타잎명은 ".BAT"로 입력해야 합니다.
다음은 MSX-DOS를 이용하여 일괄처리 프로그램을 실행시키기 전에 알아 두어야 할 사항을 열거한 것입니다.
1. "BATCH.BAT"와 같이 화일명을 BATCH로 입력할 수 없읍니다.
2. 일괄처리 화일의 실행은 화일명만을 입력하여야 하며, 타잎명은 입력시킬 필요가 없읍니다.
3. 일괄처리 프로그램의 실행중 [CTRL]+[C]를 치면 다음과 같은 메시지가 나타납니다.
   A>Terminate batch job(Y/N)?
   이때 Y 또는 y키를 치면, 아직 실행되지 않고 남은 명령들은 무시되고 MSX-DOS 프롬프트를 출력합니다. 만약, N 또는 n키를 치면 남은 명령들을 계속하여 실행하게 됩니다.
4. 일괄처리 프로그램의 실행중 디스켓을 제거하고 다른 디스켓을 넣으면 MSX-DOS는 다음 명령을 수행하기 전에, 해당 일괄처리 프로그램이 들어있는 디스켓을 요구하는 메시지를 내보냅니다.
   A>Insert disk with batch file
   and strike any key when ready
5. 일괄처리 프로그램의 맨 마지막 명령이 다른 일괄처리 프로그램의 화일명이면 첫번째 화일의 수행 후, 두번째 화일을 수행합니다.

10. AUTOEXEC.BAT 화일

MSX-DOS가 작동되면 명령어 프로세서는 MSX-DOS 디스켓에서 AUTOEXEC.BAT 화일을 찾아 자동적으로 실행하며, AUTOEXEC.BAT 화일이 없는 경우에는 날짜와 시각의 입력을 요구하는 메시지를 출력하게 됩니다.

11. 일괄처리 프로그램의 작성방법

일괄처리 프로그램은 다음과 같은 순서에 의하여 작성할 수 있읍니다.
1. .BAT 화일의 작성을 위한 명령어와 화일명을 입력시키고 [RETURN]키를 칩니다.
   A>COPY CON EXAMPLE.BAT
2. 수행하고자 하는 명령어를 입력시키고 [RETURN]키를 칩니다. 여러개의 명령어를 수행하고자 할 경우, 이 과정을 되풀이 합니다.
   A>COPY CON EXAMPLE.BAT
   DIR A:
   DIR B:
   COPY *.* B:*.AAA
   ■

3 MSX-DOS란

3. 일괄처리 프로그램의 작성이 끝나면 [CTRL]+[Z]를 누른 후 [RETURN]키를 치면 화일이 디스켓에 등록됩니다. 화일의 작성이 끝난 후, "DIR" 명령어를 이용하여 디렉토리의 내용을 보면"EXAMPLE.BAT"라는 화일이 생성되어 있음을 알 수 있읍니다.

```
A>COPY CON EXAMPLE.BAT
DIR A:
DIR B:
COPY  *.*  B:*.AAA
[CTRL]+[Z]
[RETURN]
```

4. 컴퓨터에 EXAMPLE을 입력시키면, 위의 예에서 나타낸 명령어가 순차적으로 수행됨을 확인할 수 있읍니다.

> **참고**
> - 만약 화일명이 AUTOEXEC.BAT인 경우에는 전원을 켰을 때 초기 상태에서 날짜와 시각의 입력을 생략하고 곧 .BAT 화일을 실행합니다.
> - 날짜 및 시각의 입력이 필요한 경우, 일괄처리 화일에 DATE 또는 TIME과 같은 명령을 입력하면 됩니다.

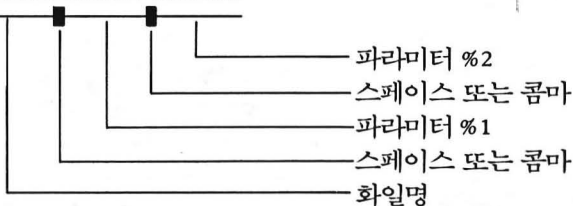
12. 가변 파라미터를 포함한 .BAT 화일

일괄처리 프로그램 작성시, 형태는 일정하나 일부 항목이 변경되는 화일인 경우에는 각 항목이 변경될 때마다 새로운 화일을 작성해야 합니다. 그러나 가변 파라미터(dummy parameters)를 이용하면 한 개의 .BAT 화일로서도 처리가 가능합니다. MSX-DOS에서는 파라미터의 값을 %0~%9까지 지정하고 있읍니다.

다음과 같은 "MYFILE.BAT"란 이름을 가진 화일을 예로 가변 파라미터를 설정하였읍니다.

```
A>COPY CON MYFILE.BAT
COPY %1.MAC %2.MAC
TYPE %2.PRN
TYPE %0.BAT
```

그림 1-4 가변 파라미터를 포함한 .BAT 화일

위와 같은 내용의 .BAT 화일의 내용을 입력시킨 후, [CTRL]+[Z]를 치면 MYFILE.BAT 화일이 만들어 집니다. 여기서 MYFILE.BAT 화일을 수행하기 위하여는, 각 파라미터에 대한 구체적인 값을 주어야 합니다.
가변 파라미터의 가장 앞에는 수행하고자 하는 .BAT 화일명을 쓰고, 각 파라미터의 순서대로 해당 값을 입력시킨 후, [RETURN]키를 칩니다.

〔실행순서〕
1. MSX-DOS는 %0에 화일명, %1에 PROG1, %2에 PROG2를 대입하여 MYFILE을 수행합니다.
2. 수행결과

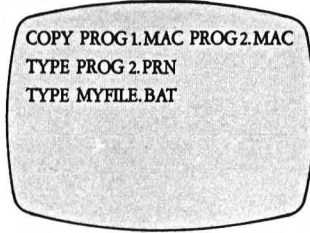



그림 1-5 .BAT 화일의 실행결과

참고 ● AUTOEXEC.BAT에서는 가변 파라미터가 포함된 명령을 수행할 수 없읍니다.
● 가변 파라미터는 스페이스 또는 콤마를 이용하여 구분하여야 합니다.

3. 가변 파라미터의 값을 바꾸면 여러 항목에 대하여 그림 1-4와 같은 일괄처리 프로그램을 수행시킬 수 있읍니다.

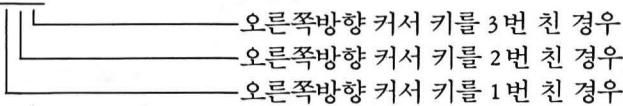
13. 에디팅 키와 제어문자의 기능

1. 에디팅 키(Editing keys)

MSX-DOS에서는 명령이 입력되면 템플리트(template)라고 하는 특수한 곳에 가장 마지막 명령을 저장하게 됩니다. 따라서 가장 마지막에 입력된 명령을 반복하여 사용할 경우에는, 필요할 때마다 되풀이하여 입력하지 않고 에디팅 키를 이용하여 편리하게 사용할 수 있읍니다.

에디팅 키로는 커서 키, SELECT, DEL, INS, BS, ESC, CLS 및 HOME 키 등이 사용됩니다. 우선 다음과 같은 명령을 입력하고 RETURN 키를 칩니다.

A >DIR B : TEST.COM

〔COPY 1〕
오른쪽방향 커서 키를 치면, 템플리트의 내용중 한 문자를 출력합니다.

〔COPY UP〕
SELECT 키를 친 후 임의의 문자를 치면, 입력된 문자의 바로 앞 열까지의 문자를 출력합니다.

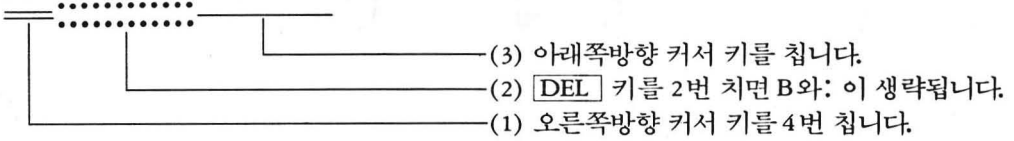
A > SELECT S RETURN
A >DIR B : TE

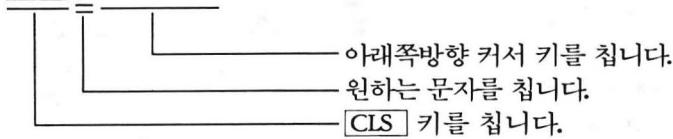
〔COPY ALL〕
아래쪽방향 커서 키를 치면, 템플리트의 내용을 모두 출력합니다.

A >DIR B : TEST.COM

3 MSX-DOS란

〔SKIP 1〕

DEL 키를 치면, 템플리트 중의 해당문자를 복사하지 않고 출력합니다.

〔SKIP UP〕

CLS 키를 친 후 임의의 문자를 치면, 입력한 문자 이전의 내용이 모두 생략됩니다.

위에서와 같은 내용을 입력하면 그 결과는 다음과 같습니다.

A >TEST.COM

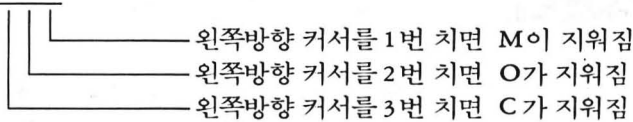
〔VOID〕

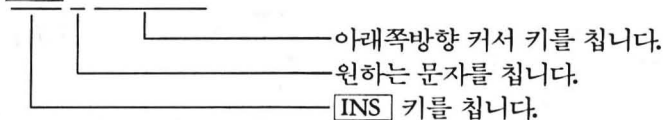
윗쪽방향 커서 키를 치면 현재 입력하고 있는 명령을 무효로 합니다. 그러나 템플리트의 내용은 그대로 남아있읍니다. 예를 들어,

A >COPY A : *.* B :

를 친후 다른 명령을 입력하거나, 명령의 형식이 잘못되어 취소해야 할 경우, 윗쪽방향 커서 키를 치면 위에서 입력한 명령이 모두 취소됩니다. (아래쪽방향 커서 키를 치면 DIR B : TEST. COM이 남아있음을 알 수 있읍니다.)

〔BS〕

왼쪽방향 커서 키를 치면 마지막으로 입력된 문자를 지웁니다.

〔INSERT〕

INS 키를 치면 현재의 커서 위치에 임의의 문자를 삽입할 수 있읍니다.

결과는 다음과 같읍니다.

A >DIR B : TEST1.COM

〔NEWLINE〕

HOME 키를 치면, 템플리트의 내용을 모두 지운 후 새로운 입력을 받아 들입니다.

A >HOME ———— HOME 키를 친다

A >TYPE MYFILE.BAT ———— 새로운 명령을 친다.

2. 제어문자의 기능(Control Character Functions)

제어문자는 [CTRL] 키와 알파베트의 일부로서 구성되며, 두 개의 키를 동시에 눌러야 합니다.

1. [CTRL]+[N] : 프린터로 출력되는 내용을 취소합니다.
2. [CTRL]+[C] : 실행중인 명령을 중지합니다.
3. [CTRL]+[H] : 명령어의 가장 마지막 문자를 삭제합니다.
4. [CTRL]+[J] : 화면의 끝에서 다음 라인으로 넘어 가고자 할 때 사용합니다.

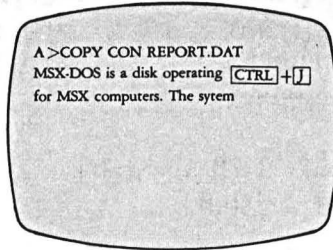



그림 1-6 [CTRL]+[J]의 기능

5. [CTRL]+[P] : 화면에 출력되는 메시지를 프린터로 출력합니다.
6. [CTRL]+[S] : 화면에 출력중인 내용을 일시 정지하며, 다시 임의의 키를 치면, 남은 내용을 계속하여 출력합니다.

14. 디스크 에러(Disk errors)

명령어나 프로그램의 수행중 디스크 에러가 발생하면 MSX-DOS는 3번 반복하여 재 수행을 시도합니다. 여기에서도 에러가 발생하면 MSX-DOS는 다음과 같은 형식을 가진 에러 메시지를 출력합니다.

〈yyy〉error 〈I/O action〉drive X
Abort, Retry, Ignor ? ■

여기에서 〈yyy〉는 에러의 종류에 따라 다음과 같은 3가지의 메시지로 출력됩니다.

Write protect
Not ready
Disk

<I/O action>은 다음과 같습니다.

reading
writing

drive X는 에러가 발생된 드라이브명을 나타냅니다.

MSX-DOS는 에러 메시지를 출력한 뒤, A, R 또는 I로 입력되는 한 개의 문자를 기다리게 됩니다. 각 문자의 역할은 다음과 같습니다.

A : Abort로서 수행중인 명령에 대한 읽기/쓰기 동작을 중지합니다.
R : Retry로서 현재 수행중인 명령을 계속하여 반복하며, 에러 발생이 없으면 명령의 수행을 마치나 또 다시 에러가 발생하면 에러 메시지를 출력합니다.
I : Ignor로서 에러의 발생을 무시하고, 명령이나 프로그램을 계속 수행합니다.

2. MSX-DOS 명령어

MSX-DOS 명령어는 다음과 같습니다.

BASIC : MSX-BASIC으로 넘어갑니다.
COPY : 지정된 화일을 복사합니다.
DATE : 현재의 날짜를 표시하고, 새로운 날짜를 입력합니다.
DEL : 지정한 화일을 삭제합니다.
DIR : 디렉토리에 있는 모든 화일에 대한 정보를 나타냅니다.
FORMAT : 데이타의 읽기와 쓰기에 필요한 형식으로 디스켓을 초기화 합니다.
MODE : 모니터 화면의 한 줄에 표시되는 최대 문자 수를 지정합니다.
PAUSE : .BAT 화일에서 키 입력을 기다립니다.
REM : .BAT 화일에서 설명문을 나타냅니다.
REN : 지정된 화일명을 바꿉니다.
TIME : 현재의 시각을 표시하고 새로운 시각을 입력합니다.
TYPE : 지정된 화일의 내용을 화면에 출력합니다.

BASIC (베이직)

형식 : BASIC[〈filespec〉]
기능 : MSX-BASIC으로 넘어간다.
설명 : MSX-DOS하에서 MSX-BASIC이나, DISK-BASIC을 실행할 경우에 사용하며 〈filespec〉에 BASIC으로 프로그램된 화일의 이름을 써 넣으면 BASIC으로 넘어간 후 해당 프로그램을 실행한다.
예 : A>BASIC DEMO.BAS

DEMO.BAS ── 화일명

COPY (카피)

형식 : COPY 〈filespec1〉 [〈filespec2〉]
기능 : 한 개 또는 그 이상의 화일을 다른 디스켓에 복사한다.
설명 : 〈filespec1〉을 〈filespec2〉로 복사한다. 그러나 동일 디스켓상에서 같은 화일명으로 복사를 할 경우에는 다음과 같은 에러 메시지를 출력한다.

File cannot be copied onto itself
0 files copied

〈filespec2〉에는 다음과 같은 3가지의 형식이 있다.

1. 〈filespec2〉가 드라이브명 만으로 된 경우 〈filespec1〉의 화일명이 해당 드라이브에 그대로 복사된다.

   A>COPY MSXDOS.SYS B:

2. 〈filespec2〉가 특정 화일명으로 구성된 경우에는 동일 드라이브상에 특정 화일로 복사한다.

   A>COPY MSXDOS.SYS MSXDOS.ABC

3. 〈filespec2〉가 드라이브 및 특정 화일명으로 주어진 경우 지정된 드라이브에 지정된 화일명으로 복사한다.

   A>COPY MSXDOS.SYS B:MSXDOS.ABC

COPY 명령으로서 화일의 연결이 가능하다. 연결방법은 COPY 명령어에 대한 부가명령으로서 "+" 기호를 사용한다. 예를 들어 MSXDOS.SYS와 COMMAND.COM 화일을 하나의 화일로 합한 후 MSXCOM.XYZ라고 하는 화일로서 B 드라이브에 기록하고자 할 경우 다음과 같은 명령을 입력시킨다.

A>COPY MSXDOS.SYS+COMMAND.COM B : MSXCOM.XYZ

여러개의 화일을 동시에 처리하고 싶을 때에는 와일드 카드를 사용할 수 있으며 다음에 그 예를 나타내었다.

A>COPY ＊. LST+ ＊.REF COMBIN.PRN

결합된 화일명과 동일한 화일명이 이미 디스켓에 들어있는 경우에는 에러로 처리된다. COPY 명령은 결합하고자 하는 화일명과 결합된 후의 화일명을 비교하여 두 화일명이 같은 경우에는 입력 화일은 건너뛰고 "Content of destination lost before COPY"라는 메시지를 출력한다. 남은 화일이 있는 경우 화일의 결합은 정상적으로 수행한다.

완성 화일이 지정되지 않은 경우에는 결합되는 화일중 가장 앞에 있는 화일명으로 결합된다. 다음은 화일 A.COM과 B.COM이 결합되어 A.COM 화일로 기록되는 예를 나타낸 것이다.

A>COPY A.COM+B.COM

화일의 결합은 보통 아스키 화일(ASCII file)로 이루어진다. 즉, 화일내에 [CTRL]+[Z]가 나타나면 화일의 끝(EOF ; End Of File)으로 해석한다. 만약 바이너리 화일을 결합하는 경우에는 스위치 /B를 이용하여야 한다. 아스키 화일과 바이너리 화일을 결합하기 위하여는 아스키 화일에는 /A를 바이너리 화일에는 /B를 사용한다. 다음의 예로서 상세한 설명을 하였다.

A>COPY A.XYZ+B.COM/B+B : C.TXT/A BIGFILE.TXT

스위치 /A, /B는 이것이 붙여지는 화일에 영향을 주고, 그 후에 다시 지정되지 않는 한, 해당 명령어 내에서 유효하게 된다. 완성 화일에 /A를 지정한 경우에는 화일의 끝을 나타내는 EOF가 써지고, /B를 지정한 경우에는 EOF가 써지지 않는다.

DATE (데이트)

형식 : DATE [〈 yy 〉—〈mm〉—〈dd〉]

기능 : 현재의 날짜를 표시하고 새로운 날짜의 입력을 받는다.

설명 : DATE를 입력하면 모니터에는 다음과 같이 현재 날짜를 출력한 뒤, 새로운 날짜의 입력을 기다린다.

Current date is Tue 87-03-17
Enter new date : ■

날짜의 변경이 필요하지 않은 경우에는 [RETURN] 키를 친다. 새로운 날짜를 입력할 경우에는 년, 월, 일의 순서로 입력시키며, "—" 또는 "/"로서 구분해야 한다.

1987-11-30
1987/11/30

MSX-DOS는 월의 크고 작음이나 윤년등을 자동적으로 구분하여 표시한다.

날짜를 잘못 기록한 경우에는 다음과 같은 메시지를 출력하고, 재 입력을 요구한다.

Invalid date
Enter new date : ■

DEL (딜리트)

형식 : DEL
기능 : 한 개 또는 그 이상의 지정된 화일을 삭제한다.
설명 : ERASE와 같이 쓸 수 있으며, 지정된 화일을 삭제한다. 와일드 카드를 이용할 수 있다.

A>DEL B : COMMAND.COM

DEL *. * 와 같이 와일드 카드를 사용한 경우에는 "Are you sure? (Y/N)" 메시지를 출력한다. 여기에서 "Y"를 치면 디스켓내의 모든 화일을 삭제하게 되며, "N"을 치면 명령이 취소된다.

A>DEL B : *. *
Are you sure? (Y/N)

DIR (디렉토리)

형식 : DIR〔〈filespec〉〕〔/P〕〔/W〕
기능 : 디렉토리에 있는 모든 화일에 대한 정보를 나타낸다.
설명 : 디렉토리에 있는 화일명, 타잎, 크기, 최후에 수정된 날짜, 시간등을 표시한다.
예 : A>DIR

현재 사용중인 드라이브의 모든 화일에 대한 디렉토리 정보를 나타낸다.

A>DIR▮B:FILENAME. *

B 드라이브의 화일중, 화일명이 FILENAME인 화일에 대하여 디렉토리 정보를 나타낸다.

A>DIR *.EXT

현재 사용중인 드라이브의 화일중 화일명과는 관계없이 타잎명이 EXT인 화일에 대한 디렉토리 정보를 나타낸다.

DIR 명령은 다음과 같은 2개의 부가명령을 가지고 있다.

〔/P〕

출력중인 화면의 내용이 한 화면에 찼을 때 일시적으로 정지하고자할 경우 사용하며, 계속 화면의 내용을 볼려면 임의의 키를 친다.

A>DIR/P
A>DIR B : *.COM/P

〔/W〕

화일명과 타잎명만을 출력한다.

A>DIR/W
A>DIR B: MYFILE. */W

FORMAT (포맽)

형식 : FORMAT

기능 : MSX-DOS가 디스켓상에서 데이타의 쓰기나 읽기 동작을 할 수 있도록 일정한 형식으로 초기화한다.

설명 : 디스켓을 처음 사용하거나 초기화 할 필요가 있을 경우 수행한다. 디스크 포맽은 다음 순서에 따라 진행된다.

1. "FORMAT"을 친후 [RETURN] 키를 친다.
2. "Drive name? (A, B)" 라는 메시지가 출력되면 포맽하고자 하는 드라이브명을 입력시킨다.
3. "Strike a key when ready" 메시지가 출력되면 임의의 키를 친다.
4. 디스켓의 포맽이 끝나면 다음과 같은 메시지를 출력한다.
   Format complete

MODE (모드)

형식 : MODE <문자 수>

기능 : 화면의 1 행에 표시할 수 있는 문자의 수를 결정한다.

설명 : 표시할 수 있는 문자의 최대 수는 80문자까지 가능하다.

예 : A>MODE 80

PAUSE (포우즈)

형식 : PAUSE [COMMENT]

기능 : 일괄처리 화일의 실행을 일시 중지한다.

설명 : PAUSE는 .BAT 화일의 수행중 명령의 수행을 잠시 중단하고자 할 때 사용한다. 이 명령은 [CTRL]+[C] 키를 제외한 임의의 키를 치면 명령을 재개한다. PAUSE 명령을 실행할 때는 다음과 같은 메시지를 출력한 뒤 .BAT 화일의 수행을 일시 중지한다.

Strike a key when ready

PAUSE 명령에서는 [COMMENT]를 줄 수 있으며 이는 화일의 사용자에게 어떤 메시지를 전달하고자 할 때 사용한다.

예 : PAUSE

PAUSE Now format routine. Exchange your diskette
[COMMENT]

REM (리마아크)

형식 : REM [COMMENT]

기능 : 일괄처리 화일의 실행중 REM 명령을 만나면 이 명령의 뒤에 나오는 [COMMENT]를 출력한다.

설명 : REM은 그 뒤에 나오는 [COMMENT]의 표시 이외에는 .BAT의 수행에는 어떠한 영향도 주지 않는다. REM 명령과 [COMMENT] 사이에는 스페이스(SPACE), 탭(TAB) 또는 콤마(,) 등으로 구분하여야 한다.

예 : REM This is TEST program for floppy disk drive
스페이스 [COMMENT]

REN (리네임)

형식 : REN 〈filespec〉 〈filename〉

기능 : 〈filespec〉 으로 정의된 화일명을 새로운 화일로 변경한다.

설명 : RENAME도 사용할 수 있으며, REN과 같은 기능을 수행한다. 화일의 지정에는 와일드 카드를 사용할 수 있으며 화일이 변경되면 본래의 화일은 삭제된다. 변경전의 화일명과 변경후의 화일명은 스페이스, 탭 또는 콤마 등으로 구분하며, 변경된 후의 화일명이 이미 디스켓에 들어 있는 경우에는 다음과 같은 에러를 출력한다.

Rename error

예 : A>REN COMMAND.COM ABCDE.XYZ

; COMMAND.COM 화일을 ABCDE.XYZ 화일로 변경한다.

A>REN ＊.LST ＊.PRN

; 타입명이 LST인 화일을 타입명이 PRN인 화일로 변경한다.

A>REN B:ABCXYZ ?D?E?F

; 드라이브B의 화일중 ABCXYZ을 ADCEYF로 변경한다.

TIME (타임)

형식 : TIME〔〈hh〉〔:〈mm〉〔:〈dd〉〕〕〕

기능 : 현재의 시각을 표시하고 새로운 시각의 입력을 받는다.

설명 : TIME을 입력하면 모니터에는 다음과 같이 현재 시각을 출력한 뒤, 새로운 시각의 입력을 기다린다.

Current time is 11:30:20.35a
Enter new date : ■

시각의 변경이 필요하지 않은 경우에는 [RETURN] 키를 친다. 새로운 시각을 입력할 경우에는 시, 분, 초의 순서로 입력시키며, " : "로서 구분해야 한다.

17:53:12
5:53:12p

MSX-DOS는 오전과 오후 등을 구분할 수 있으며, 위의 보기에서와 같이 오후 시각은 p를 입력하며, 오전 시각은 a를 입력하면 된다.

시각을 잘못 기록한 경우에는 다음과 같은 메시지를 출력하고, 재 기입을 요구한다.

Invalid time
Enter new time : ■

TYPE (타이프)

형식 : TYPE 〈filespec〉

기능 : 화일의 내용을 수정없이 보고자할 때 사용한다.

설명 : 〈filespec〉에서 정의된 화일의 내용을 화면상에 출력한다. 그러나 모든 화일이 가능한 것은 아니며 아스키 화일의 출력만이 가능하다. TYPE 명령은 다음과 같은 특성을 가지고 있다.

- TYPE 뒤에는 반드시 화일명이 지정되어야 한다.
- TYPE 명령에는 와일드 카드를 사용할 수 없으며 만약 사용한 경우에는 에러 메시지를 표시한다.
- 화일의 내용을 모니터와 프린터로 동시에 출력하고자 하면 [CTRL]+[P]를 누른다. 프린터로의 출력을 정지하고자 할 경우[CTRL]+[P]를 다시 한번 누른다.

1. A>TYPE B : MYFILE.BAT

; B드라이브에 있는 MYFILE.BAT의 내용이 모니터로 출력된다.

```
COPY %1.MAC %2.MAC
TYPE %2.PRN
TYPE %0.BAT
```

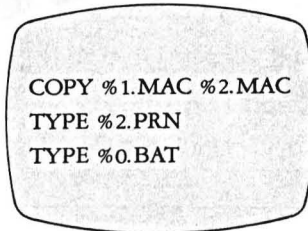



그림 2-1 화일 내용의 출력 (그림 1-4 참고)

2. [CTRL]+[P]
A>TYPE B: MYFILE. BAT

; 현재 사용중인 드라이브명, 명령어 및 화일의 내용을 모니터와 프린터로 출력한다. 이 때, 프린터가 연결되어 있지 않은 경우에는 에러가 발생한다.

```
A>TYPE B: MYFILE. BAT
COPE %1.MAC %2.MAC
TYPE %2.PRN
TYPE %0.BAT
```

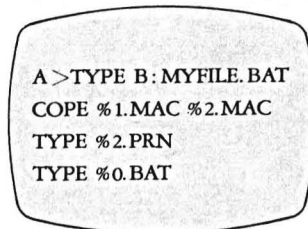
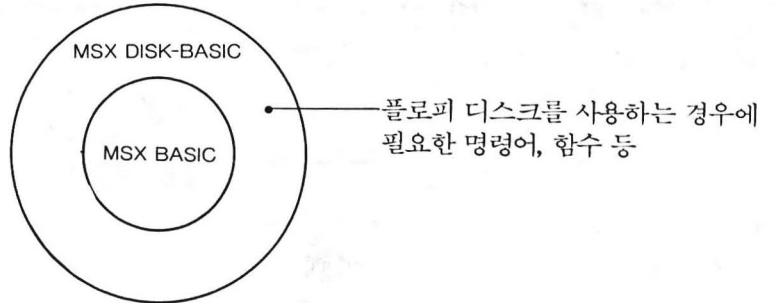



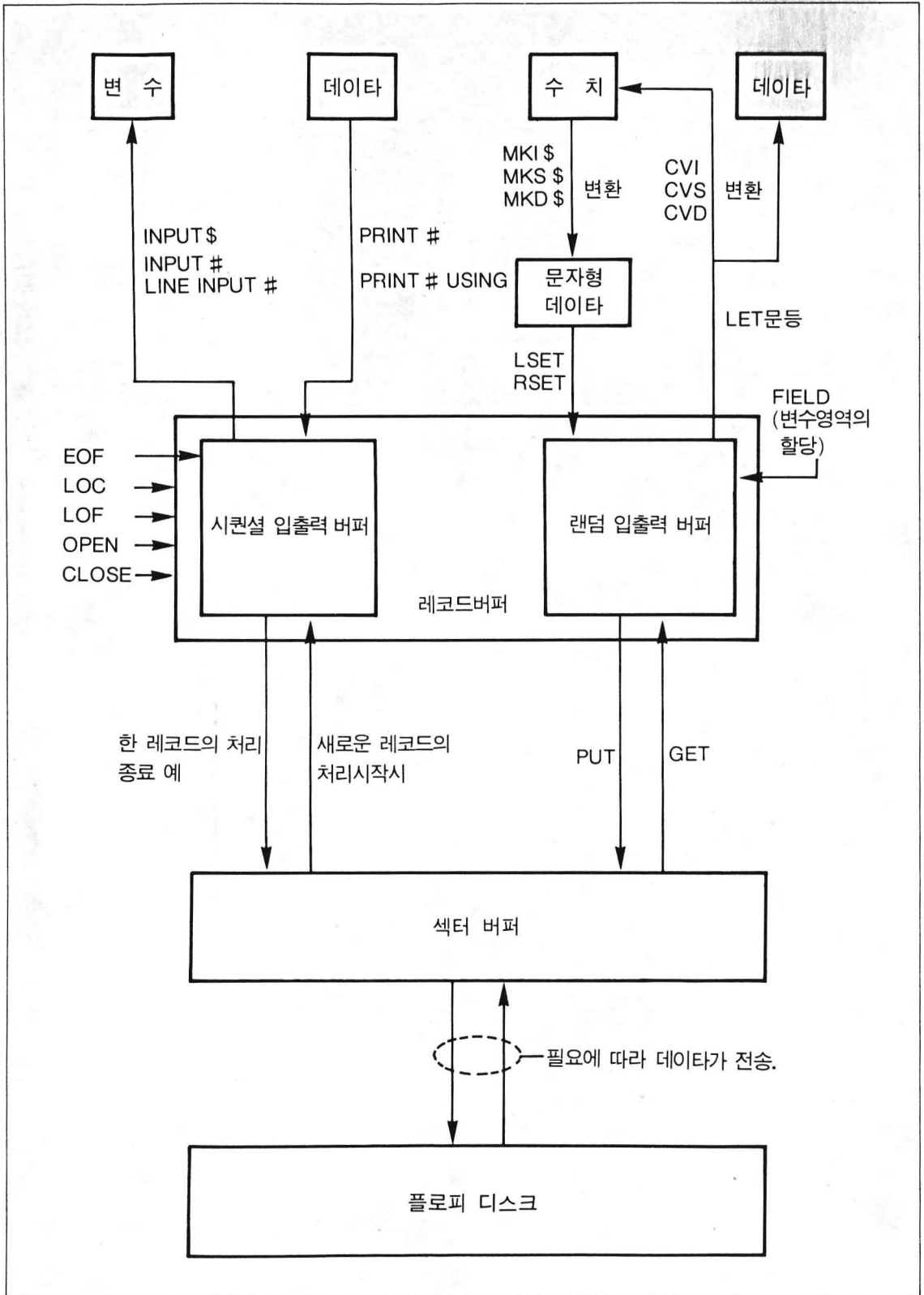
그림 2-2 화일 내용의 출력

3. DISK-BASIC에 대하여

MSX DISK-BASIC

MSX DISK-BASIC은 MSX 컴퓨터가 가지고 있는 데이타를 화일로서 디스켓에 기록하거나, 디스켓의 화일을 컴퓨터의 메모리에 읽어내기 위하여 사용하는 명령과 함수(function)를 포함한 BASIC 언어 전체를 말합니다.

즉 MSX 베이직을 포함한 플로피 디스크를 사용하기 위한 언어입니다. 디스크 베이직의 명령, 문, 함수에는 여러가지 종류가 있으나, 자세한 내용은 「베이직 설명서」편을 참조해 주십시오.

디스크 베이직의 명령문들을 그림으로 정리했으니 참조하여 주십시오.

종	류	명령, 문, 함수
플로피 디스크에 써넣기, 읽기	써넣기	SAVE, BSAVE, DSKO$
	읽기	LOAD, BLOAD, DSKI$, RUN
화일 조작 관계	데이타의 전송	INPUT #, INPUT$ LINE INPUT #, PRINT # PRINT # USING, PUT, GET LSET, RSET
	버퍼의 설정 상태의 조사	OPEN, CLOSE, FIELD EOF, LOC, LOF
	데이타의 변환	MKI$, MKS$, MKD$ CVI, CVS, CVD
유틸리티		FORMAT, COPY, FILES KILL, MERGE, DSKF NAME, SYSTEM

변 수
데이타
수 치
데이타
MKI $
MKS $
MKD $
변환
CVI
CVS
CVD
변환
INPUT $
INPUT #
LINE INPUT #
PRINT #
PRINT # USING
문자형
데이타
LET문등
LSET
RSET
FIELD
(변수영역의
할당)
EOF
LOC
LOF
OPEN
CLOSE
시퀀셜 입출력 버퍼
랜덤 입출력 버퍼
레코드버퍼
한 레코드의 처리
종료 예
새로운 레코드의
처리시작시
PUT
GET
섹터 버퍼
필요에 따라 데이타가 전송.
플로피 디스크

4

베이직 설명서

1. 베이직(BASIC)이란?

우리들이 「컴퓨터를 사용한다」는 것은, 컴퓨터에게 어떤 일을 시키는 것입니다. 컴퓨터에게 「무엇인가」를 시키기 위해서는, 컴퓨터가 이해할 수 있는 언어를 사용하여, 그것에 대한 명령을 컴퓨터에게 알려주어야 합니다. 이처럼 컴퓨터를 위한 언어에는 많은 종류가 있지만, 그중에서도 우리들이 쉽게 이해할 수 있는 언어는 **베이직**이라는 언어입니다. 베이직(BASIC)이란, 「Beginners All-purpose Symbolic Instruction Code」의 준말로써, 초보자를 위해 알기쉽게 되어있으며, 컴퓨터 언어중에서 가장 흔히 사용되고 있읍니다. 그러므로 베이직을 사용하여 MSX가 이해할 수 있는 명령을 기억시켜 놓으면, 누구라도 MSX를 생각한대로 사용할 수 있을 것입니다.

BASIC의 동작 모드

X-II에 전원을 넣으면, 화면에는 「Ok」라는 문자가 표시됩니다. 이 상태를 베이직 모드라 하며, 베이직 명령을 입력하는 것이 가능하게 됩니다. 베이직 명령은 **직접모드**와 **간접모드**(프로그램모드)의 두가지로 실행됩니다.

1. 직접모드

베이직의 명령문을 키보드로 입력한 후, [RETURN] 을 누르면, 바로 실행됩니다. 이것을 **직접모드에서의 실행**이라고 합니다. 이 모드를 사용하여 아래와 같은 계산을 해봅시다.

```
PRINT 10+7-2
```

이때, PRINT는 「화면에 데이타를 표시하라」라는 명령이고, 「PRINT 10+7-2」 전체를 **명령문**이라고 합니다. 이경우, 명령문이 실행되면, 「10+7-2」의 계산 결과인 「15」가 화면에 표시됨을 알 수 있읍니다. 이와 같이, 직접모드에서는 베이직 명령을 입력한 후 [RETURN] 을 누르면, 바로 실행됩니다.

2. 간접모드(프로그램모드)

베이직 명령문 앞에 번호를 붙여 입력하면, 이 명령문은 프로그램내의 행으로 되어 번호와 함께 컴퓨터에 기억됩니다. 이 번호를 **행번호**라 하며, 행번호가 붙은 명령문의 집합을 **프로그램**이라고 합니다. 프로그램은 RUN이라는 명령으로 실행되며, 이것을 **간접모드에서의 실행**이라고 합니다.

프로그램을 만드는 방법

프로그램이란, 명령의 집합입니다. 행번호가 붙은 베이직 명령문을 입력하고, [RETURN] 을 치면, 그 행은 직접모드에서처럼 바로 실행되지 않고, 프로그램내의 한 행으로 취급되어, 컴퓨터의 메모리에 기억됩니다.
행번호는 0~65529의 정수로, 메모리에 기억되는 순서를 표시하며, 또 완성된 프로그램은 행번호가 작은 것부터 차례로 실행됩니다.

주의 새로운 프로그램을 입력하기 전에 반드시 NEW(뉴우)명령을 입력하여, 메모리에 남아있는 프로그램을 지워 주십시오.

```
10 A=123
20 PRINT A
30 END
```

위의 프로그램을 실행하면, 명령은 행번호가 작은 순으로 실행되어, 화면에는 「123」이 표시됩니다.

참고

베이직의 명령은 알파벳의 대문자 또는 소문자로 입력하여도 관계없읍니다. 컴퓨터의 내부에서는 모두 대문자로 바뀌어 처리됩니다. 그러나 인용부호(")로 묶여진 문자는 입력한 그대로 기억됩니다. 예를들면,

```
10 print "abc"
20 PRINT "ABC"
30 end
```

위와같이 대문자와 소문자를 각각 입력하여도, LIST 명령을 해보면 아래와 같이 표시됩니다.

```
10 PRINT "abc"
20 PRINT "ABC"
30 END
```

명령은 소문자인 것이 대문자로 바뀌어졌으나, 인용부호로 묶여진 문자는 원래 그대로 기억되고 있읍니다.

프로그램의 수정

프로그램을 고치고 싶을 때는 아래와 같이 합니다.

① 행의 추가

추가할 행번호에 내용을 붙여 입력합니다. 예를들어, 프로그램의 20행과 30행 사이에 명령을 추가하려면, 그 사이의 번호(예를들면, 23이나 25등)에 새로운 명령을 붙여 입력하고, [RETURN] 을 누릅니다.

② 행의 삭제

삭제하고 싶은 행의 행번호를 입력하고, [RETURN] 을 치면, 그 행은 삭제됩니다. 연속된 여러 행들을 한꺼번에 삭제할 때는 DELETE 명령을 사용합니다. (명령어편의 DELETE명령 참조.)

③ 행의 내용을 수정

수정하고 싶은 행의 행번호에 새로운 내용을 입력하고, [RETURN] 을 누르면, 이전의 행의 내용은 새로운 내용으로 변합니다. 또 LIST 명령을 사용하여 프로그램을 화면에 표시한 후, 커서를 수정하고 싶은 행으로 이동하여, 내용을 수정하고 [RETURN] 을 누릅니다.

④ 행번호를 바꿈

행번호들을 바꿔 붙이고 싶을 때는 RENUM 명령을 사용합니다. 자세한 것은 명령어편의 RENUM 명령을 참조하십시오.

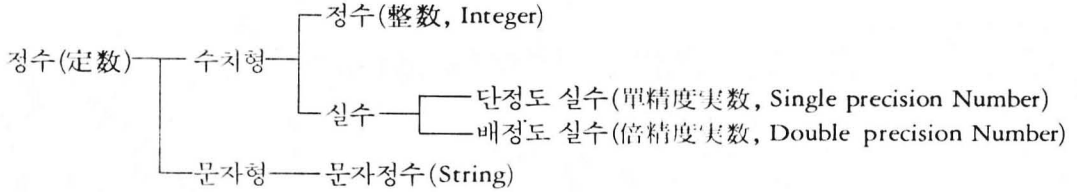
이와같이 프로그램을 수정, 추가, 삭제하는 것을 에디팅(편집)이라 하며, 에디팅을 효과적으로 하기 위해서는 [INS], [BS], [DEL], 커서키 등을 이용합니다.

용어의 해설

- **문** : 명령문이라고 하며, 컴퓨터에게 주는 명령의 내용을 나타냅니다.
- **행** : 컴퓨터에게 명령을 줄 때의 단위이며, 행번호부터 [RETURN] 을 칠 때까지 입력한 것을 말합니다. 한 행은 255문자 이내이며, 한 개이상의 문을 포함할 때는 :(콜론)으로 구분해야 합니다.
- **행번호** : 행을 여러개 모아 프로그램을 만들 경우에 행의 앞에 붙이는 번호입니다.
- **데이타** : 일반적으로 여러가지 종류의 정보를 가리키며, 베이직에서는 사용자가 지정하는 문자나 숫자를 말합니다.
  예를들어, 「A=1」이라는 것은 A라는 변수에 1이라는 값을 대입한다는 뜻으로, 이경우 1을 데이타라고 합니다.

4 베이직 설명서

정수(定數)

베이직에서 취급하는 정수(定數), 즉 데이타는 다음과 같습니다.

1. 문자정수(文字定數)

인용부호(")로 묶여진, 255자 이내의 문자, 부호, 공백, 숫자 등으로 구성되는 문자열이며, 산술연산에는 사용될 수 없읍니다.

예) "THANK YOU"
"3.1415926"
"TEL 999-9999"

2. 수치정수(数値定数)

수치는 산술연산을 할 수 있는 데이타이며, 크게 실수형과 정수형으로 나누어져 있읍니다.

1. 정수(整数)

정수는 −32768∼+32767의 범위이며, 다음 4가지 형식으로 사용됩니다.

*** 10진수(Decimal)형식**

음의 정수인 경우, 반드시 마이너스(—)부호를 붙여 주어야 하지만, 양의 정수인 경우, 플러스(+)부호는 생략해도 좋읍니다. 또, −32768∼+32767의 실수 뒤에 %를 붙이면, 소숫점 이하는 버리고 정수형으로 됩니다.

예) 0
—123
3875.6%→3875

*** 8진수(Octal)형식**

0부터 7까지의 숫자 앞에 &O를 붙여 표시합니다. 범위는 &O0∼&O177777이며, &O177777∼&O100000는 음수로, 10진수의 −1∼−32768에 대응합니다.

예) &O12345 →10진수로는 5349
&O177777→10진수로는 −1

*** 16진수(Hexdecimal)형식**

16진수(0∼9, A∼F) 앞에 &H를 붙여 표시합니다. 범위는 &H0∼&HFFFF이며, &HFFFF∼&H8000는 음수로, 10진수의 −1∼−32768에 대응합니다.

예) &H100 →10진수로는 256
&HFFFF→10진수로는 −1

* 2진수(Binary)형식

0과 1로된 2진수 앞에 &H를 붙여 표시한 것입니다. 범위는 &B0~&B1111111111111111이며, &B1111111111111111~&B1000000000000000는 음수로 10진수의 −1~−32768에 대응합니다.

예) &B01100100 → 10진수로는 100
&B1010101010101010 → 10진수로는 −21846

> **주의** 8진수 형식, 16진수 형식, 2진수 형식으로 입력된 숫자는 PRINT나 LPRINT 명령으로 출력하면, 10진수 형식으로 표시됩니다. 그러므로 10진수 이외의 형식으로 출력할 때는 각각 OCT$, HEX$, BIN$ 등을 사용하여 문자열로 출력됩니다.

2. 실수형

실수는 표시되는 숫자에 의해 단정도 실수와 배정도 실수로 나누어집니다. MSX베이직에서 실수의 연산은 전부 배정도 실수로 변하여 행해지고, 결과도 배정도 실수로 얻읍니다. 그러나 식에서 단정도 실수를 선언하면, 결과는 단정도 실수로 됩니다.

* 단정도(單精度, Single Precision)실수형

단정도 실수로 표시되는 수치는 −9.99999E+62~+9.99999E+62이며, 지수의 범위는 E+62~E−64입니다.
단정도 실수에는 다음 세가지 형식이 있읍니다.

- 6개 이내의 숫자로 구성된 수치(일곱번째 자리에서 반올림 합니다.)
- 지수기호 E를 사용한 부동소숫점 형식
- 형선언문자 「!」를 붙인 실수

예) 12345.6
1.986E02 → 1.986×10^2
34567.55! → 34567.6

* 배정도(倍精度, Double Precision)실수형

배정도 실수로 표시되는 수치는 −9.9999999999999E+62~+9.9999999999999E+62이며, 지수의 범위는 E+62~E−64입니다.
배정도 실수에는 다음 세가지 형식이 있읍니다.

- 14개 이내의 숫자로 구성된 수치(15번째 자리에서 반올림 합니다)
- 지수기호 E 또는 D를 사용한 부동소숫점 형식
- 형선언문자 「#」을 붙인 실수

예) 12345678.95789325 → 12345678.957893
−1.09432D−06 → -1.09432×10^{-6}
0.3141592653E01 → $0.3141592653 \times 10^{1}$
36 #

수치형 데이타는 필요(메모리의 양 및 실행속도 등)에 따라 형을 변환시킬 수 있으나, 문자형과 수치형 사이에서는 변환시킬 수 없읍니다. 형변환은 다음 규칙에 따라 행해집니다.

- 수치를 다른 형의 수치변수에 대입하는 경우, 수치는 대입 전에 변수의 형으로 변합니다.
- 정도가 다른 수치의 연산인 경우, 정도가 낮은 쪽이 높은 쪽의 형으로 변환되어 연산이 실행됩니다.
- 논리연산의 경우, 사용되는 수치는 모두 정수형으로 변환되며, 결과도 정수형으로 됩니다.
- 실수가 정수로 변환되는 경우는 소숫점 이하는 버려지고, 그 결과가 정수형의 범위(—32768~32767)를 넘으면, 에러가 발생됩니다.
- 배정도 실수형 변수를 단정도 실수형 변수에 대입하면, 변수의 값은·유효자리 6자리의 값으로 되고, 배정도 실수형의 일곱번째 자리에서 반올림됩니다.

변수(Variables)

변수는 프로그램 중에서 사용하는 데이타를 바꿔 넣을 수 있는 것으로, 영숫자로된 이름(변수명)이 붙읍니다.
변수에 값을 넣기 위해서는 LET문이나 INPUT 명령등을 이용하며, 어떤 경우에도 변수의 형은 그것에 대입되는 데이타의 형과 일치하여야 합니다.
변수에 값을 대입하기 전에는, 숫자변수는 0, 문자변수는 널(Null : 문자가 들어있지 않은 상태)로 되어 있읍니다.

1. 변수의 형

변수의 형은 변수명 끝에 형선언문자를 붙여서 구별합니다. 형선언문자에는 다음 네가지가 있읍니다.

% : 정수형	# : 배정도형
! : 단정도형	$: 문자형

이 형선언문자를 사용하면, 변수명이 같아도 다른 변수로 취급됩니다.(예를들어, A$, A ! 등)
그리고 형선언문자를 생략하면, 배정도형으로 취급됩니다.
변수의 형을 선언하는 또다른 방법은 프로그램 중에서 DEFINT/SNG/DBL/STR(디파인 인티저/싱글/더블/스트링)명령을 사용하는 것입니다.

용어의 설명

- **변　　수** : 그 자체가 값을 표시하는 것이 아니고, 반드시 대응된 값이 대입되어 식의 일부로 되는 것입니다.
- **변 수 명** : MSX 베이직에서의 변수명은 두 문자까지의 알파벳이나 숫자입니다. 두 문자 이상을 표시할 수 있으나 컴퓨터에 기억될 때는 앞의 두 문자만 기억됩니다. 맨 앞의 문자는 반드시 영문자이어야 하며, 변수명에 예약어를 포함해서도 안됩니다. (예약어에 대해서는 자료편의 「예약어표」 참조)

  예) 변수명이 틀린 경우
  3AX　(최초의 문자가 숫자입니다.)
  B/3　(「/」는 사용할 수 없읍니다.)
  COST (COS는 예약어이므로 사용할 수 없읍니다.)

  변수명이 올바른 경우
  BB, X1, XYZ(XY만 기억됨), SUM(SU만 기억됨)

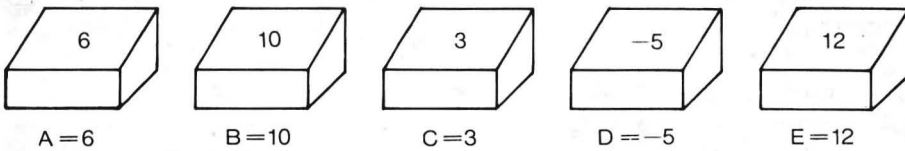
● **형선언문자**: 정수나 변수의 형을 표시하는 기호로, 수치나 변수명의 마지막에 붙여서 사용하고 있읍니다.
형선언문자를 사용하면, 메모리내에서 데이타가 차지하는 영역을 줄일 수 있읍니다.

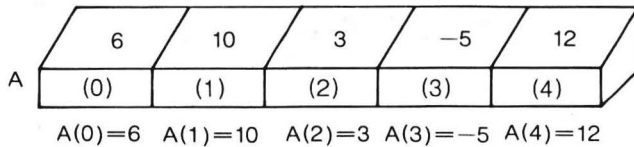
● 다음은 각 변수가 사용하는 메모리의 크기를 표시한 것입니다.

정수형변수 : 5바이트→변수형(1)+변수명(2)+정수형수치(2)바이트
단정도형변수 : 7바이트→변수형(1)+변수명(2)+단정도형수치(4)바이트
배정도형변수 : 11바이트→변수형(1)+변수명(2)+배정도형수치(8)바이트
문자형변수 : 6+(문자수)바이트→변수형(1)+변수명(2)+문자열의 문자갯수(1)+문자열저장번지(2)+(문자수)바이트

2. 배열변수(Array Variables)

배열변수는 같은 성질(즉, 같은 변수형)을 가진 복수 데이타의 집합으로 X(1), X(2)와 같이 변수명 뒤에 있는 괄호에 번호를 붙여서 사용합니다. 괄호의 번호를 **첨자**라 하고, 배열중의 몇번째 요소인가를 표시합니다. (예를들어, X(2)는 X라는 배열변수의 두번째 요소라는 의미가 됩니다.)
배열변수는 많은 데이타를 변수에 대입하는 경우에 사용합니다. 예를들어, 변수가 100개이면, 변수명도 100개가 필요하지만, A(1)～A(100)과 같이 첨자만 변화시키면 되므로, 결국 변수명은 A하나로 충분합니다.

＊ 다른 변수명을 사용할 경우

각각의 상자(변수)에 하나의 데이타가 들어가므로 다섯개의 상자(변수)가 필요합니다.

＊ 배열변수를 사용할 경우

하나의 상자(배열변수)가 5개의 작은 상자(변수)로 나누어져 있고, 작은 상자(변수)마다 데이타가 들어가므로, 변수명은 하나만 필요합니다.

배열변수의 첨자는 0부터 시작합니다.

예) A(10) → 수치배열변수 → 요소의 수 11
B$(4) → 문자배열변수 → 요소의 수 5
F$(3, 3) → 2차원의 문자배열변수 → 요소의 수 $4 \times 4 = 16$
N(2, 5, 3) → 3차원의 수치배열변수 → 요소의 수 $3 \times 6 \times 4 = 72$

배열변수의 차원

위에서 설명한 배열변수는 X(1)과 같이 첨자가 하나였지만, 예)에서 표시한 F$(3, 3)이나 N(2, 5, 3)과 같이 복수의 첨자를 가진 배열변수도 사용할 수 있읍니다.
복수의 첨자를 사용할 경우에, 배열의 차원은 첨자의 갯수에 대응됩니다.
예를들어,

DIM D(2, 3)

는 아래에 표시된 2차원 배열입니다.

D(0, 0)	D(0, 1)	D(0, 2)	D(0, 3)
D(1, 0)	D(1, 1)	D(1, 2)	D(1, 3)
D(2, 0)	D(2, 1)	D(2, 2)	D(2, 3)

마찬가지로 DIM Y(2, 3, 4)는 3차원 배열입니다.

이와 같이 배열의 차원은 몇 개의 첨자를 지정하는가에 따라 결정됩니다.
차원은, 이론적으로는 255차원까지 사용할 수 있으나, 실제로는 메모리의 용량때문에 제한됩니다.

3. 시스템 변수(System Variables)

MSX 베이직에서 베이직 자체의 목적을 위한 변수를 시스템 변수라고 합니다. 시스템 변수에는 다음과 같은 아홉가지가 있읍니다. (명령어편의 각 명령 참조)

TIME (타임) : 시간을 계산합니다. 1/60초마다 값이 증가하는 변수입니다.
ERL (에러라인) : 에러가 발생될 때, 발생된 행번호를 얻읍니다.
ERR (에러) : 에러가 발생될 때, 에러의 원인을 표시하는 에러코드를 얻읍니다.
BASE (베이스) : 화면 출력에 관계되는 테이블의 번지를 얻읍니다.
SPRITE$ (스프라이트 스트링) : 지정된 스프라이트의 패턴을 만듭니다.
VDP (비디오 프로세서) : 비디오 프로세서의 레지스터에 데이타를 써 넣거나 읽어냅니다.
CSRLIN (커서라인) : 텍스트 화면에서 커서의 수직위치를 조사합니다.
POS (포지션) : 텍스트 화면에서 커서의 수평위치를 조사합니다.
LPOS (라인포지션) : 라인 프린터의 헤드의 위치를 조사합니다.

식과 연산

식이란, 연산(계산)의 문법을 표시하는 것으로, 정수나 변수를 연산기호(계산에 사용하는 특수기호)로 연결한 것입니다. 식의 연산결과는 수치 또는 문자열로 되며, 단순한 수치나 문자만으로 된 것도 식이라고 합니다.

예) "BASIC"
10+3/5
2
A+B/C−D

1. 산술연산

다음은 산술연산 기호를 연산의 우선 순위로 나타낸 것입니다.

연 산 기 호	연 산	예 문
∧	지 수	2∧3→2^3
−	마이너스부호	−2
*, /	곱셈, 나눗셈	5*3, 5/3
₩	몫	16₩5→3
MOD	나머지	23 MOD 5→3
+, −	덧셈, 뺄셈	5+3, 5−3

참고

- **₩** : 나눗셈의 몫을 구하며, 수치가 실수이면, 연산이 실행되기 전에 소숫점 이하는 버립니다.
  예) 16₩5 ——————→ 3
  23.75₩5.2 → 23₩5 → 4
- **MOD** : 나눗셈의 나머지를 구하며, 수치가 실수이면, 연산이 실행되기 전에 소숫점 이하는 버립니다.
  예) 14 MOD 3 ——————→ 2
  23.75 MOD 5.3 → 23 MOD 5 → 3
- **0으로 나눌 때** : 숫자를 0으로 나누면, "Division by Zero"(디비젼바이제로)라는 에러 메시지가 표시되고, 명령 대기상태로 돌아갑니다.
- **오버플로우(Overflow)** : 연산이나 변수의 수치를 대입할 때, 그 결과가 허용되는 범위를 초과했을 경우에는, "Overflow"라는 에러 메시지가 표시되고, 명령대기 상태로 돌아갑니다.

4 베이직 설명서

2. 관계연산

관계연산은 두 개의 수치 또는 문자열을 비교할 때 사용합니다. 관계연산의 결과는 참(−1), 거짓(0)으로 나타나며, IF(이프)명령 등으로 프로그램의 흐름을 변화시키는데에 사용합니다. 관계연산 기호에는 다음과 같은 것이 있읍니다.

관계 연산기호	내 용	예
=	같다	X=Y
<>, ><	같지 않다	X<>Y, X><Y
<	작다	X<Y
>	크다	X>Y
<=, =<	작거나 같다	X<=Y, X=<Y
>=, =>	크거나 같다	X>=Y, X=>Y

〈예〉 IF X=0 THEN 100 (만약 X가 0이면, 행번호 100을 실행합니다.)
IF A+B<>0 THEN X=A+B(만약 A+B가 0이 아니면, X에 A+B의 값을 대입합니다.)

3. 논리연산

논리연산은 복수의 조건을 조사한다든지, 또는 비트조작이나 부울연산을 행하는데 사용합니다. 논리연산의 결과는 비트마다 0 또는 1로 나타납니다. 각 논리연산의 내용은 다음과 같읍니다.

NOT(부정)

X	NOT X
1	0
0	1

AND(논리곱)

X	Y	X AND Y
1	1	1
1	0	0
0	1	0
0	0	0

OR(논리합)

X	Y	X OR Y
1	1	1
1	0	1
0	1	1
0	0	0

XOR(배타적논리합)

X	Y	X XOR Y
1	1	0
1	0	1
0	1	1
0	0	0

EQV(같음)

X	Y	X EQV Y
1	1	1
1	0	0
0	1	0
0	0	1

IMP(포함)

X	Y	X IMP Y
1	1	1
1	0	0
0	1	1
0	0	1

IF명령으로 이와 같은 논리연산 기호를 사용하여, 여러 개의 조건을 판단할 수 있읍니다.

(예) IF X<0 OR 99<X THEN 100(X가 음수이거나 99보다 크면, 행번호 100을 실행합니다.)
IF 0<X AND X<100 THEN X=0(X가 양수이고 100보다 작으면, X에 0을 대입합니다.)
IF NOT(A=0) THEN 20(A가 0이 아니면, 행번호 20을 실행합니다.)

4. 함수

함수란, 지정된 수치를 가지고 지정된 연산을 행하여, 그 결과를 나타내는 것입니다.
MSX 베이직에서 사용되는 함수는 다음과 같습니다.

- 수치함수 : SIN(삼각함수의 사인), SQR(평방근) 등
- 문자함수 : CHR$(캐랙터 스트링), LEFT$(레프트 스트링) 등
- 사용자 정의 함수 : 사용자가 임의로 정의할 수 있는 함수로, DEF FN(디파인 펑션 네임) 함수 등.

각각의 함수에 대해서는 명령어편을 참조하십시오.

5. 문자열연산

1. 문자열의 연결

문자열은 연산기호 「+」를 사용하여 연결시킬 수 있습니다. 예를들면,
A$="봄이" : B$="왔다"이면, A$+B$="봄이 왔다"입니다.

2. 문자열의 비교

문자열의 비교도 수치처럼 관계연산 기호를 사용하여, 각 문자의 문자코드를 서로 비교하는 것입니다.
각각의 문자열의 처음부터 한 문자씩 비교하며, 양쪽이 전부 같으면 그 두개의 문자열은 같다고 판단되고, 한 문자라도 틀리면 그 문자의 문자코드가 큰쪽의 문자열이 크게 됩니다.
문자열의 비교에서는 공백도 의미를 가지므로, 주의하십시오.

(예)		
	"AB"<"BC"	: A(문자코드 : 41)가 B(문자코드 : 42)보다 작읍니다.
	"CAT"<"CATS"	: 문자코드가 같을 때는 문자열의 길이가 긴 쪽이 큽니다.
	"PEN ">"PEN"	: 공백(문자코드 : 32)도 의미를 가집니다.
	"cm">"CM"	: 소문자 c(문자코드 : 63)가 대문자 C(문자코드 : 43)보다 큽니다.

6. 연산의 우선 순위

여러가지 연산기호의 우선 순위는 다음과 같읍니다.

1) 괄호로 묶여진 것
2) 함수
3) 지수(∧)
4) 마이너스부호(−)
5) *, /
6) ₩
7) MOD
8) +, −
9) 관계연산자(=, <등)
10) NOT
11) AND
12) OR
13) XOR
14) IMP
15) EQV

연산의 우선순위를 변경하고자 할 때는 괄호를 사용하여, 괄호속의 연산자를 다른 연산자보다 먼저 실행하게 합니다.

2. 명령어 보는법

◎ 명령은 찾아보기 쉽게 알파벳순으로 나열하였으며, 그 기능을 간단히 설명하였읍니다.
한편 초보자도 곧 익숙해지도록 명령 읽는 법을 한글로 표시하였읍니다.

◎ 문형식 : 명령을 사용하기 위한 방법을 표시합니다.

● 영문자로 표시된 항목은 그대로 입력합니다.
베이직 명령은 소문자와 대문자를 구별하지 않으므로, 바꿔써도 관계없읍니다. 그렇지만 시스템내에서는 모두 대문자로 취급합니다.

(예) LOCATE→LACATE ; 명령을 잘못 표기했읍니다.
PRINT→Print ; 서로 같은 명령입니다.
PEEK→PeeK ; 서로 같은 명령입니다.

● 명령이 아닌 문자열(“ ”로 둘러싸인 문자열)은 대문자와 소문자를 구별해야 합니다.

(예) “MSX”와 “mSx”는 서로 같지 않습니다.

● 〈 〉으로 묶여진 항목은 사용자(USER)가 지정합니다.

(예) POKE〈번지〉, 〈데이타〉
→사용자가 지정

● 〔 〕안에 있는 항목은 생략하거나, 어느 하나를 선택하여 쓸수 있읍니다. 생략되면 그 명령 실행전에 지정된 값이 적용됩니다.

(예) CLEAR〔〈문자열 영역의 크기〉〔, 〈메모리의 상한〉〕〕
〔 〕전체를 생략할 수 있으며, 또한 그중 하나만 선택할 수도 있읍니다.
CLEAR
CLEAR 〈문자열 영역의 크기〉
CLEAR 〈문자열 영역의 크기〉, 〈메모리의 상한〉

● 쉼표(,)로 구분된 여러개의 파라메터(PARAMETER)에 있어서, 생략가능한 경우에는 다음과 같은 서식으로 표시합니다.

(예) CIRCLE〔STEP〕(X, Y), 〈반지름〉〔, 〈색번호〉〕〔, 〈시작 각도〉〕〔, 〈끝각도〉〕〔, 〈비율〉〕
이 경우 〔 〕으로 묶여진 파라메터는 생략할 수 있읍니다. 어떤 파라메터와 이후의 파라메터를 전부 생략하는 경우는 쉼표(,)도 포함해서 생략할 수 있으나, 뒤에 계속되는 파라메터중 어느 하나라도 지정할 경우에는 그 이전에 있는 쉼표는 전부 지정해 주어야 합니다.
CIRCLE(100, 96), 50, 1 ; 색번호 이후의 파라메터는 생략
CIRCLE(150, 50), 50, , , , 0.4 ; 비율을 지정해 주었기 때문에 비율앞의 쉼표를 생략할 수 없읍니다.

● 다음 예에서와 같이 파라메터뒤에 쉼표(,) 혹은 세미콜론(;)을 지정하는 경우, 쉼표와 세미콜론을 포함해서 생략할 수 있읍니다.

(예) INPUT〔〈"프롬프트문"〉 |;| 〕〈변수〉〔,〈변수〉…〕

● 〈 〉, 〔 〕를 제외한 기호, 즉 쉼표(,), 세미콜론(;), 괄호(( )), 하이픈(—), 등호(=) 등은 지시하는 위치에 반드시 입력해야 합니다.

● 생략기호 「……」는 한 행이 허락하는 범위 (255문자) 내에서만 가능하다는 것을 나타냅니다.

(예) DATA 〈정해진 수〉〔,〈정해진 수〉……〕
데이타는 한 행에 들어가는 범위내에서 쉼표를 구분하여 몇개라도 지정할 수 있읍니다.

◎ **설명** : 명령의 사용법이나 기능 또는 그것에 관한 주의할 점을 설명합니다. 비슷한 내용의 명령에서 사용되는 파라메터에 대해서는 해설편 등에서 설명하였읍니다.

(예) 화일명

◎ 설명 다음에 이해를 돕기 위하여 간단한 예문을 들고, 그에 대한 설명을 했읍니다.

(예) GET DATE A$; 문자변수 A$에 현재의 날짜가 대입됩니다.

◎ **참조** : 관련이 있는 명령문이나 설명을 참조할 수 있도록 했읍니다.
각각의 명령을 서로 비교하여, 이해하는데 도움이 되도록 하십시오.

3. 베이직에서 사용하는 특수기호

MSX 베이직에서 사용되는 문자는 영문자(대문자와 소문자), 숫자, 한글, 특수기호, 그래픽문자 등으로 구성되어 있습니다.
이중에서 베이직 명령어 또는 베이직 프로그램 작성에 특별한 의미를 가지고 있는 특수기호에 대하여 설명합니다.

- **.(마침표)** : 현재 베이직이 가리키고 있는 행번호를 표시합니다.
  프로그램을 수정하거나 행을 삽입했을 때, 또는 프로그램을 실행하여 에러가 발생할 때, 그 행을 가리킵니다.
  한편 LIST, RENUM, DELETE 등에서 행번호대신 사용할 수 있읍니다.
  (예) LIST. ; 수정이나 삽입한 행 등이 표시됩니다.

- **—(마이너스)** : 수치의 범위를 지정할 때 사용합니다.
  (예) LIST 100—200 ; 프로그램중의 행번호 100부터 200까지의 List가 표시됩니다.

- **:(콜론)** : 여러개의 문을 구분할 때 사용합니다.
  (예) A=B+C : PRINT A

- **,(쉼표)** : PRINT, INPUT, DATA문 등에서 파라메터나 수치를 여러개 지정하는 경우, 그것을 구분할 때 사용합니다. 「PRINT」 명령에서의 쉼표는 데이타를 14자리마다 구분하여 표시합니다.
  (예) DATA A, B, C
  PRINT 1, 2

- **;(세미콜론)** : PRINT문 등에서 수치나 문자열을 구분할 때 사용합니다. 「PRINT」 명령에서 사용할 경우, 데이타는 앞의 데이타에 연결되어 표시됩니다.
  (예) PRINT A; B; C

- **'(홑따옴표)** : REM(리마아크)명령 대신에 사용합니다.
  (예) 'TEST PROGRAM→REM TEST PROGRAM

- **?(물음표)** : PRINT 명령 대신에 사용합니다. ?라고 입력해도 컴퓨터에서는 PRINT라고 기억됩니다.
  (예) ? 25 * 12→PRINT 25 * 12

- **(공백)** : 프로그램을 알아보기 쉽도록 문중에 적당한 공백을 줄 수 있읍니다.
  단, 명령, 문, 함수, 시스템 변수 등의 예약어 중에는 공백을 넣을 수 없으며, 문자정수 중의 공백은 문자로서 의미를 가지게 됩니다. 명령문 사이의 공백은 프로그램을 실행할 때는 영향을 주지 않지만, 컴퓨터의 메모리에 저장되기 때문에 그만큼 메모리를 많이 차지하게 됩니다.

4. 명령어 해설

ABS (앱스)

절대값(ABSOLUTE VALUE)을 얻는다.

● 문형식 : ABS (〈수식〉)

● 설　명 : 〈수식〉에 대한 절대값이므로, 얻어진 결과는 항상 0이거나 양수로써 배정도 실수형이다.

A＝ABS(－1.2)　　; A에는 －1.2의 절대값인 1.2가 대입된다.

● 참　조 : SGN

ASC (아스키)

문자코드(ASCII CODE)를 얻는다.

● 문형식 : ASC (〈문자식〉)

● 설　명 : ○ 〈문자식〉이 여러개의 문자로 되어 있는 경우에는, 첫 문자의 문자코드를 얻는다.
○ 인용부호(")는 문자로 취급되지 않으며, 〈문자식〉이 널 스트링(Null String)인 경우는 "Illegal function call" (일레갈 펑션 콜) 에러가 발생한다.
○ 문자와 문자코드의 대응관계는 자료편의 「문자코드」를 참조하십시오.

A＝ASC("A")　　; A에는 「A」의 아스키코드 65가 대입된다.
B＝ASC("CDE")　; B에는 「CDE」의 첫 문자인, C의 아스키코드 67이 대입된다.

● 참　조 : CHR$

ATN (아크탄젠트)

아크탄젠트(ARC TANGENT)의 값을 얻는다.

● 문형식 : ATN (〈수식〉)

● 설　명 : ○ 〈수식〉은 어떠한 형의 수치라도 좋다.
○ 얻어진 값은 배정도 실수형이고, 단위는 라디안(RADIAN)이다.
○ 라디안과 각도의 관계에 대한 자세한 설명은 SIN을 참조하십시오.

A＝ATN(1)　; A에는 1의 아크탄젠트 값인 0.78539816339745 라디안이 대입된다.

● 참　조 : TAN

AUTO (오토)

행번호가 자동으로 발생된다.

● 문형식 : AUTO〔〈행번호〉〕〔,〈증가〉〕

● 설　명 : ○ 〈행번호〉는 0～65529의 정수이며, 〈증가〉는 1～65529의 정수이다.
○ 〈행번호〉를 생략하면 0, 〈증가〉를 생략하면 10이 지정되고, 〈행번호〉와 〈증가〉를 동시에 생략하면 각각 10이 지정된다.
○ 프로그램중에 이미 사용된 행번호가 발생할 경우에는 행번호 뒤에 아스터리스크(＊)가 표시된다. 이 때, 프로그램을 입력하고 RETURN 을 누르면, 그 행에 새로운 내용이 들어가게 되나, 아무 것도 입력하지 않고 RETURN 을 누르면, 그 행은 먼저 있던 내용을 그대로 보존한다.
○ AUTO 기능을 중지하고 싶을 때는 CTRL + C 또는 CTRL + STOP 을 누른다. 이 때, 최후에 표시된 행번호의 행은 기억되지 않는다.

```
AUTO        ; 행번호는 10, 20, 30, …순으로 된다.
AUTO 100    ; 행번호는 100, 110, 120, …순으로 된다.
AUTO ,20    ; 행번호는 0, 20, 40, 60, …순으로 된다.
AUTO 20,5   ; 행번호는 20, 25, 30, 35, …순으로 된다.
```

BASE (베이스)

화면 출력에 관계되는 테이블의 번지를 얻는다.

● 문형식 : BASE (〈수식〉)

● 설　명 : ○ 〈수식〉의 범위는 VDP 레지스터 번호인 0～44이다.
○ VDP로 V9938이 사용되고 있으며, BASE 함수를 사용할 때는 V9938을 잘 이해한 후 사용해야 한다. 만약 그렇지 않으면, 시스템이 오동작할 경우도 있다.
○ 이 시스템(System) 변수에는 값을 대입할 수 없다.

```
A=BASE(10) ;  A에는 고해상도 그래픽모드의 명칭 테이블의 번지가 대입된다.
```

● 참　조 : VDP

BEEP (비이프)

소리를 낸다.

● 설　명 : ○ BEEP는 스피커를 약 0.04초간 울린다.
○ 제어코드 7과 같은 명령이므로, PRINT CHR$(7)을 실행하여도 BEEP 명령과 같은 결과를 얻을 수 있다.
○ 미리 SET BEEP 명령을 사용하여, BEEP 음의 음색을 여러가지로 할 수 있다.

● 참　조 : SET BEEP, 자료편의 「제어코드표」

BIN$ (바이너리 스트링)

2진수의 문자열을 만든다.

● 문형식 : BIN$(〈수식〉)

● 설　명 : ○ 〈수식〉의 범위는 −32768부터 65535까지이다.
　○ 결과는 16자리까지 나타낼 수 있으나, 맨 앞부터 계속되는 0은 표시되지 않는다.

A$=BIN$(5)　;A$에는 10진수 5에 대한 2진수 형식의 문자열 "101"이 대입된다.
B$=BIN$(70)　;B$에는 10진수 70에 대한 2진수 형식의 문자열 "1000110"이 대입된다.

● 참　조 : VAL, OCT$, HEX$

BLOAD (비로드)

기계어 프로그램을 읽어들인다.

● 문형식 : BLOAD "[〈장치명:〉] [〈화일명〉]" |[,R] / [,S]| [, 〈옵셋〉]

● 설　명 : ○ 〈장치명:〉은 외부 기억장치를 나타내며, 데이타 레코더는 "CAS:", 디스크드라이브는 "A:" 또는 "B:" 등이다.
　데이타 레코더만 연결된 경우에는 "CAS:"를 생략할 수 있다. 데이타 레코더와 디스크 드라이브가 연결된 상태에서 〈장치명:〉을 생략하면, 현재 사용중인 드라이브가 지정된다.
　○ 〈화일명〉에 대해서는 해설편의 「화일」을 참조하십시오.
　○ R옵션을 지정하면 프로그램을 읽어들인 후, BSAVE로 지정해 놓은 실행번지부터 즉시 실행된다.
　○ ,S는 디스크 드라이브에서만 가능하며, 이것을 지정하면 읽어들인 데이타는 비디오메모리에 기록된다. ,R과 ,S는 동시에 지정할 수 없다.
　○ 〈옵셋〉을 지정하면 BSAVE로 지정된 시작번지에 〈옵셋〉을 더한 메모리 번지부터 써 넣는다.

BLOAD "CAS:TEST"　;「TEST」라는 화일명의 기계어 프로그램을 데이타 레코더로부터 읽어들인다.
BLOAD "TEST"　;데이타 레코더와 디스크 드라이버가 함께 연결된 경우, 현재 사용중인 드라이브로부터 「TEST」 프로그램을 읽어들인다.
BLOAD "TEST", R　;「TEST」 프로그램을 읽어들인 후, 바로 실행한다.
BLOAD "TEST", &H00A0;「TEST」 프로그램을 BSAVE 명령으로 지정한 시작번지에 &H00A0를 더한 번지부터 읽어들인다.
BLOAD "B:TEST"　;B드라이브로부터 「TEST」 프로그램을 읽어들인다.
BLOAD "A:TEST.BAS"　;화일명 「TEST」에 타잎명 「BAS」가 붙은 프로그램을 읽어들인다.
BLOAD "A:TEST", S　;「TEST」 프로그램을 읽어들여 비디오메모리에 기록한다.

● 참　조 : BSAVE, CLOAD, LOAD

BSAVE (비세이브)

기계어 프로그램을 보존한다.

● **문형식** : BSAVE"〔〈장치명:〉〕〈화일명〉",〈시작번지〉,〈끝번지〉 |〔,〈실행번지〉〕 / 〔,S〕|

● **설　명** : ○ 〈시작번지〉부터 〈끝번지〉까지의 메모리에 있는 프로그램을 기계어로써 지정한 화일에 보존한다.

○ 〈화일명〉은 생략할 수 없으며, 〈화일명〉에 대해서는 해설편의 「화일」을 참조하십시오.

○ 〈장치명〉은 BLOAD 명령을 참조하십시오.

○ 〈시작번지〉, 〈끝번지〉, 〈실행번지〉의 범위는 −32768이상 65535이하의 정수이다. 〈실행번지〉를 지정하면, R 옵션이 붙은 BLOAD 명령문으로 기계어 프로그램을 읽어들였을 때, 지정한 번지부터 실행이 시작되나, 생략하면 〈시작번지〉가 〈실행번지〉로 간주된다.

○ ,S는 디스크 드라이브에서만 가능하며, 비디오메모리의 내용을 보존할 때 사용한다.

```
BSAVE "CAS:TEST", &H9000, &H9FFF        ;시작번지(&H9000)부터 끝번지(&H9FFF)까지의
                                         메모리에 있는 프로그램을 기계어로써
                                         데이타 레코더에 「TEST」라는 이름으로 보존하고,
                                         BLOAD 명령으로 읽어들일 때 실행번지는
                                         &H9000이다.
BSAVE "TEST", &H9000, &H9FFF, &H9100;실행번지는 &H9100이다.
BSAVE "B:TEST", &H2020, &H2040, S       ;비디오메모리의 &H2020부터 &H2040까지에 있는
                                         내용을 B드라이브에 「TEST」라는 이름으로 보존한다.
```

● **참　조** : BLOAD, CSAVE, SAVE

CALL (콜)

확장문을 호출한다.

● **문형식** : CALL〈확장문 이름〉〔(〈인수〉〔,〈인수〉…〕)〕

● **설　명** : 베이직명령이 확장될 경우, 확장문을 호출한다.
CALL 대신에 _(언더바, under bar)를 사용할 수 있다.

```
CALL HANON = _HANON
```

CALL ADJUST (콜어드저스트)

전자펜을 사용할 수 있게 한다.

● **설　명** : ○ 전자펜을 사용가능하게 해주며, 화면의 옵셀 좌표를 설정한다.

○ 이 명령은 전자펜을 위한 PAD함수, PAD(20)~PAD(23)와 함께 사용된다.

CALL FORMAT (콜포맷)

디스켓을 초기화 한다.

●설 명 : 이 명령을 실행하면, 다음과 같은 메뉴가 나타난다.

Drive name ? (A, B)

드라이브명을 선택하면, 다시 다음과 같은 메세지가 나타난다.

Strike a key when ready

이 때, 새 디스켓을 디스크 드라이브에 넣고, 키를 누르면 디스크 드라이브가 작동을 시작하고, 디스켓이 초기화(포맷) 된다.

주의 프로그램이 기록되어 있는 디스켓을 포맬하면 디스켓안에 있는 모든 화일들이 지워지므로 주의하여야 한다.
또, 새 디스켓은 사용하기전에 반드시 포맬을 해야만 한다.

CALL MEMINI (콜메모리이니셜)

메모리디스크를 초기화 한다.

●문형식 : CALL MEMINI〔(〈메모리 디스크의 크기〉)〕

●설 명 : ○ 메모리디스크내의 모든 내용을 지우고, 사용할 메모리의 영역을 지정한다.
○ 메모리디스크를 사용하기 위해서는, 먼저 이 명령을 실행해야 한다.

CALL MEMINI(&H3FFF) ; 메모리디스크를 초기화하고, 사용할 수 있는 영역을 &H0에서 &H3FFF까지로 한다.

●참 조 : 해설편의 「메모리 디스크」

CALL MEMINI(0) (콜메모리이니셜제로)

메모리디스크의 기능이 해제된다.

●설 명 : 메모리디스크에 기록되어 있던 모든 프로그램과 데이타가 지워진다.

CALL MFILES (콜메모리파일스)

메모리디스크에 있는 모든 화일 이름을 나타낸다.

●설 명 : 메모리디스크에 있는 화일명과 남아있는 메모리디스크의 용량이 표시된다.

CALL MKILL (콜메모리킬)

메모리디스크내의 지정된 화일을 지운다.

● **문형식 :** CALL MKILL (〈"화일명"〉)

● **설　명 :** 〈화일명〉에 대한 자세한 설명은 해설편의「화일」을 참조하십시오.

CALL MKILL("TEST")　;메모리 디스크내에 있는「TEST」라는 화일을 지운다.

● **참　조 :** 해설편의「메모리 디스크」

CALL MNAME (콜메모리네임)

메모리디스크에 있는 화일명을 변경한다.

● **문형식 :** CALL MNAME (〈"구화일명"〉 AS 〈"신화일명"〉)

● **설　명 :** ○ 〈구화일명〉을 〈신화일명〉으로 변경시킨다.
○ 〈구화일명〉이 메모리에 없거나, 〈신화일명〉이 이미 메모리디스크에 있는 경우에는 에러가 발생한다.
○ 〈화일명〉에 대한 자세한 설명은 해설편의「화일」을 참조하십시오.

CALL MNAME("TEST" AS "SAMPLE")　;메모리디스크에 있는「TEST」라는 화일 이름을「SAMPLE」이라는 이름으로 변경한다.

● **참　조 :** 해설편의「메모리 디스크」,「화일」

CALL SYSTEM (콜시스템)

베이직 상태에서 MSX-DOS로 되돌아간다.

● **설　명 :** ○ 이 명령은 MSX-DOS에서 베이직으로 왔을때만 사용할 수 있다.
○ 모든 화일은 닫히고 메모리안에 있던 프로그램과 데이타는 지워진다.

CDBL (컨버트더블)

수치를 배정도 실수로 변환한다.

● **문형식 :** CDBL (〈수식〉)

● **설　명 :** 〈수식〉의 값을 배정도 실수로 변환한다.

A=CDBL(1/3)　;A에는 1/3배정도 실수값인 0.33333333333333이 대입된다.

CHR$ (캐렉터스트링)

문자코드에 대응되는 문자를 얻는다.

● **문형식 :** CHR$ (〈수식〉)

● **설　명 :** 〈수식〉은 0～255의 범위이며, 또한, 수치가 제어코드(0～31)로 정의되어 있으면 문자가 표시되는 것이 아니라 그 코드가 갖는 기능이 실행된다.

A$=CHR$(65)　;A$에는 문자코드 65에 대응하는 문자 A가 대입된다.

● **참　조 :** ASC, 자료편의「문자코드표」

CINT (컨버트인테져)

수치를 정수(整數)로 변환한다.

● 문형식 : CINT(〈수식〉)

● 설 명 : 결과는 정수이므로 −32768∼32767의 범위를 넘으면 "Overflow" (오버플로우) 에러가 발생한다.

A=CINT(123.45678) ;A에는 소숫점 이하를 버린 123이 대입된다.

CIRCLE (서어클)

그래픽 화면에 원, 타원, 부채꼴을 그린다.

● 문형식 : CIRCLE |(X, Y) / STEP(X, Y)|, 〈반지름〉 〔, 〈색번호〉〕 〔, 〈시작각도〉〕 〔, 〈끝각도〉〕 〔, 〈비율〉〕
(단, X=수평좌표, Y=수직좌표)

● 설 명 : ○ 이 명령은 그래픽 모드에서만 사용할 수 있다. 따라서 SCREEN 문으로 미리 화면을 그래픽 모드로 지정해 두어야 한다.
○ (X, Y)는 중점을 나타내는 것으로, X는 0∼255 (단, SCREEN 6, 7에서는 511까지), Y는 0∼191 (단, SCREEN 5이상에서는 211까지)의 범위이다.
○ STEP(X, Y)를 사용하여, 마지막 참조점의 상대좌표로 중점을 지정할 수 있다. (LINE 명령 참조)
CIRCLE 명령을 실행하면, 원의 중점(X, Y)가 마지막 참조점이 된다.
○ 중점의 좌표를 중심으로 하여, 〈반지름〉으로 지정하는 크기의 원을 그린다.
○ 〈색번호〉를 지정하면, 지정한 색으로 원이 그려지고, 생략하면 이전에 COLOR 명령으로 지정한 표시색으로 된다.
○ 〈시작각도〉와 〈끝각도〉를 지정하면, 그 범위의 원호가 그려진다. 한편, 시작각도와 끝각도가 음수(−)이면, 부채꼴이 그려진다.
각도의 단위는 라디안(RADIAN)이며, 생략하면 각각 0과 π이다.
○ 〈시작각도〉와 〈끝각도〉가 −2π부터 2π의 범위 안에 없으면 "Illegal function call" (일레갈 펑션 콜) 에러가 발생한다.
○ 〈비율〉은 (수직방향의 반지름)/(수평방향의 반지름)으로 지정하며, 생략하면 비율은 1이다. 비율이 1보다 작으면 수평방향으로 긴 타원이 되고, 1보다 크면 수직방향으로 긴 타원이 된다.

```
10 SCREEN 2                                          ;화면을 그래픽 모드로 지정한다.
20 CIRCLE(100,96),50,1                               ;중점(100,96)과 반지름 50의 원을
                                                      검은색으로 그린다.
30 CIRCLE(200,140),50,15,4/5*3.14,2/5*3.14           ;시작각도를 4/5×3.14로 하고, 끝각도를
                                                      2/5×3.14로 한 하얀색 호를 그린다.
40 CIRCLE(160,130),50,2,−2/5*3.14,−5/6*3.14;시작각도를 −2/5×3.14로 하고, 끝각도를
                                                      −5/6×3.14로 한 녹색의 부채꼴을 그린다.
50 CIRCLE(150,50),50,10,,,0.4                        ;비율을 0.4로한 수평으로 긴 타원을
                                                      노란색으로 그린다.
60 CIRCLE(50,80),50,6,,,1.9                          ;비율을 1.9로한 수직으로 긴 타원을
                                                      빨간색으로 그린다.
70 GOTO 70                                           ;그래픽 모드로 계속 머물러 있도록
                                                      지정한 것이다.
```

● 참 조 : SCREEN, COLOR

4 베이직 설명서

CLEAR (클리어)

모든 변수를 초기화하며, 사용할 메모리 영역을 지정한다.

● 문형식 : CLEAR [〈문자열 영역의 크기〉 [, 〈메모리의 상한〉]]

● 설 명 : ○ 모든 변수를 초기화한다. 즉, 수치변수는 0으로 하고, 문자변수는 널스트링(Null String)이 된다.
○ 문자변수의 내용을 저장하여 놓는 〈문자열 영역의 크기〉를 지정하고, 〈메모리의 상한〉은 &H82BC～&HF380 범위내에서 지정할 수 있다.
○ 〈메모리의 상한〉을 지정할 때는 〈문자열 영역의 크기〉를 생략할 수 없다.

CLEAR 300, &HD000 ; 문자열 영역의 크기를 300바이트로 하고 베이직이 사용할 수 있는 메모리의 상한을 &HD000로 설정한다.

CLOAD/CLOAD ? (카세트로드/카세트로드?)

데이타 레코더에서 프로그램을 읽어들인다. /읽어들인 프로그램과 테이프에 있는 프로그램을 비교한다.

● 문형식 : CLOAD [?] [〈"화일명"〉]

● 설 명 : ○ CLOAD는 데이타 레코더에서 〈화일명〉으로 지정한 프로그램을 찾아내서 읽어들이는 명령이다.
○ 지정한 화일을 테이프에서 찾고 있는 도중에 다른 화일이 발견되면, 「Skip : 발견한 화일명」을 화면에 표시하면서 계속 찾아나가고, 지정한 화일을 찾으면 「Found : 화일명」을 표시한 후 로드를 시작한다.
○ CLOAD ? 명령은 보통 CSAVE 명령을 실행한 후에 기록이 바르게 되었는가를 확인할 때 사용한다.
○ 화일명은 생략할 수 있으며, 생략되면 CLOAD, CLOAD ? 가 실행되어 처음으로 찾은 프로그램(CSAVE로 된 것)을 읽어들인다.

CLOAD "TEST" ; 데이타 레코더에 CSAVE로 기록된 「TEST」라는 프로그램을 찾아 로드한다.
CLOAD ? "TEST" ; 데이타 레코더에 있는 「TEST」라는 프로그램이 현재 메모리상에 있는 프로그램과 같은가를 조사하여 같으면 「Ok」가 표시되고, 다르면 "Verify error" (베리파이에러)가 표시된다.

> 주의 CLOAD명령을 실행하면, 그때 메모리상에 있던 프로그램이 없어지므로 주의하십시오.
> CLOAD로는 SAVE명령으로 저장된 프로그램을 읽을 수가 없다.

● 참 조 : CSAVE, LOAD, BLOAD

CLOSE (클로스)

열려있는 화일버퍼를 닫는다.

● **문형식** : CLOSE〔〔#〕〈화일버퍼번호〉〔,〔#〕〈화일버퍼번호〉〕……〕

● **설 명** : ○ 여러개의 화일버퍼번호를 쉼표(,)로 연결하여 한꺼번에 닫을 수도 있다.
또 화일버퍼번호를 생략하면 열려있는 모든 화일이 닫힌다.
○ 화일버퍼를 열린 상태로 놓으면 화일버퍼중의 데이타가 파손되는 일이 있으므로 화일의 사용이 끝나면 반드시 CLOSE 명령을 실행해야 한다.
○ END, NEW, CLEAR 등의 명령을 실행하면 자동적으로 화일이 닫히나 STOP 명령은 화일을 닫을 수 없다.

```
10 OPEN "CRT:" FOR OUTPUT AS #1 ;텍스트화면을 위한 화일번호를 1로 한다.
20 PRINT #1, "COMPUTER"         ;화일번호#1에 「COMPUTER」를 출력한다.
30 CLOSE #1                     ;화일번호#1을 닫는다.
```

● **참 조** : OPEN, END, NEW, CLEAR

CLS (클리어스크린)

화면을 지운다.

● **설 명** : ○ 기능키의 내용과 스프라이트를 제외하고, 화면에 표시되어 있는 모든 것을 지운다.
○ 그래픽화면에서 COLOR 명령문을 실행할 때는 반드시 CLS 명령을 실행한다.
만약 CLS 명령을 실행하지 않으면, COLOR 명령으로 지정한 바탕색대로 되지 않는다.
○ 텍스트 화면에서 CLS가 실행되면 커서는 좌측 위로 이동한다.

● **참 조** : COLOR

COLOR (칼라)

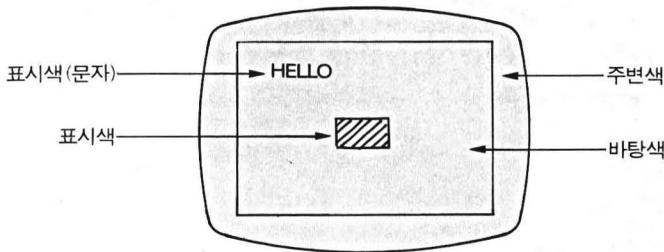
화면의 색을 지정한다.

● **문형식** : COLOR〔〈표시색번호〉〕〔, 〈바탕색번호〉〕〔, 〈주변색번호〉〕

● **설 명** : ○ 색번호는 0~15이며, 이 색번호에 대한 색상은 고정된 것이 아니고,
실제로는 칼라 빠레트 번호이므로 칼라 빠레트의 내용을 변경시킴으로써 색번호에 대한 색상을 바꿀 수 있다. 그러나 초기 상태에서는 색번호와 색상이 아래와 같이 대응한다.

0: 투명(주변색)	4: 진한 청색	8: 적색	12: 진한 녹색
1: 검정색	5: 밝은 청색	9: 밝은 적색	13: 보라색
2: 녹색	6: 진한 적색	10: 진한 황색	14: 회색
3: 밝은 녹색	7: 하늘색	11: 밝은 황색	15: 흰색

○ SCREEN 6에서의 색번호는 0~3이며, 그 이상(4~15)의 값은 4로 나눈 나머지가 색번호로 된다. 즉 색번호 5이면 1, 7이면 3, 12이면 0을 색번호로 취한다.
한편, 이 색번호는 색번호 참조 테이블에 2비트씩 반복해서 보존되므로,
SCREEN 모드가 바뀌면 다른 색상이 표시된다.
예) 1 (색번호) →01 (2비트) →0101 (4비트) →5 (색번호)
2 →10 →1010 →10

○ SCREEN 8에서는 칼라 빠레트를 사용하지 않고, 직접 색번호의 수치(0~255)로 색상을 만든다. 즉 이 수치를 2진수로 하여, 상위 3비트는 녹색의 밝기,
다음 3비트는 적색의 밝기, 항상 상위 비트를 1로하는 하위 3비트는 청색의 밝기를 표시하며, 이것들이 혼합되어 색상이 만들어진다.
예) 색번호 137은 2진수로「10001001」이다.
이것은 녹색(100), 적색(010), 청색(101)의 밝기가 혼합된 색이다.

○ 〈표시색번호〉는 화면에 문자나 점, 선을 나타낼 때 사용되는 색이다.
단, 그래픽 명령에서는 각각의 명령으로 점, 선 등에 색상 지정을 할 수 있다.
○ 〈바탕색번호〉는 화면의 바탕이 되는 색이다. 텍스트모드에서는 이 명령을 실행하면, 바로 색상이 바뀌지만, 그래픽모드에서는 이 명령을 실행한 후
SCREEN이나 CLS명령을 실행해야 바뀐다.
○ 〈주변색번호〉는 화면의 테두리로써, 베이직에서는 사용하지 않는 부분이다.
○ 전원을 넣은 후, 아무 것도 지정하지 않은 상태에서의 표시색은 15 (흰색), 바탕색은 4 (진한 청색), 주변색은 7 (하늘색)이다.

COLOR 1, 15, 8 ; 칼라 빠레트가 초기 상태이면, 표시색을 검정색으로, 바탕색을 흰색으로, 주변색을 적색으로 한다.

● **참 조** : COLOR=, COLOR=NEW

COLOR= (칼라=)

칼라빠레트에 적, 녹, 청색을 혼합한다.

● 문형식 : COLOR=(〈칼라빠레트번호〉, 〈적색의 밝기〉, 〈녹색의 밝기〉, 〈청색의 밝기〉)

● 설 명 : ○ 〈칼라빠레트번호〉는 색번호에 대응하며, 0~15의 범위이나, SCREEN 6에서는 0~3이다.
○ SCREEN 8에서는 이 명령은 의미가 없다. (이 모드에서는 칼라빠레트를 사용하지 않는다.)
○ 적, 녹, 청색의 밝기는 각각 0~7의 8단계이며, 그 값이 클수록 밝은 색을 내며, 이 3색이 혼합되어 색번호의 색상으로 된다.
○ 초기상태에서는 베이직이 칼라빠레트의 값을 지정하여 색상을 얻는다. 이것에 대해서는 COLOR 명령을 참조하십시오.

COLOR=(1, 0, 7, 7) ; 칼라빠레트번호 1에 녹색과 청색의 밝기를 최대로 한 색을 대입한다.

● 참 조 : COLOR, SCREEN, CLS

COLOR=NEW (칼라=뉴우)

칼라빠레트의 값을 초기 상태로 한다.

● 문형식 : COLOR〔=NEW〕

● 설 명 : ○임의로 되어 있는 칼라빠레트의 값을 초기 상태의 값으로 한다.
○초기상태에서의 칼라빠레트 번호와 색상의 대응 관계는 COLOR 명령을 참조하십시오.

COLOR=NEW ; 빠레트를 사용하여 혼합해 놓은 색들을 무시하고 전원을 켰을때와 같은 상태로 되돌린다.

● 참 조 : COLOR, COLOR=

COLOR=RESTORE (칼라=리스토어)

비디오 메모리(VRAM)에 있는 칼라 참조테이블의 값을 칼라 빠레트에 옮긴다.

● 설 명 : ○ 베이직은 COLOR=명령 등으로 칼라빠레트의 색을 지정할 때마다 그 값을 칼라 참조테이블에 저장한다.
○ BSAVE된 칼라 참조테이블의 내용을 BLOAD한 후, 칼라빠레트에 그 내용을 대입 해주는 명령이다.
○ 이 명령을 실행하지 않으면, 초기상태의 색번호의 색이 표시된다.

● 참 조 : BSAVE, BLOAD

4

베이직 설명서

COLOR SPRITE$ (칼라 스프라이트 스트링)

스프라이트의 각 수평라인마다 색을 지정한다.

● 문형식 : COLOR SPRITE$ (〈면번호〉) = 〈문자식〉

● 설　명 : ○ 스프라이트의 색 지정을 하지 않은 라인은 COLOR문에서 지정한 표시색 또는 PUT SPRITE문에서 지정한 색으로 된다.

○ SCREEN 4～8에서만 유효하다.

○ 〈수식〉은 0～255이며, 그 값의 2진수 표현으로, 하위 4비트는 색번호를 나타내며, 상위 4비트는 PUT SPRITE문으로 스프라이트를 화면에 표시할 때, 다음과 같은 의미를 가진다.

B7 : 1이면 32화소 왼쪽으로 이동되어 표시된다.
B6 : 1이면 스프라이트 번호가 하나 작은 스프라이트 면에 의해서 색상, 표시, 이동 등의 영향을 받으며 충돌은 검출되지 않는다.
B5 : 1이면 충돌이 검출되지 않는다.
B4 : 사용하지 않는다.

○ SCREEN 8에서는 스프라이트의 색이 다음 16색으로 고정된다.

0 : 검정색	4 : 진한 녹색	8 : 살색	12 : 녹색
1 : 진한 청색	5 : 진한 하늘색	9 : 청색	13 : 하늘색
2 : 진한 적색	6 : 진한 노란색	10 : 적색	14 : 노란색
3 : 진한 보라색	7 : 회색	11 : 보라색	15 : 흰색

COLOR SPRITE$(0) = CHR$(1) + CHR$(7) ; 칼라빠레트가 초기치라면 스프라이트 면번호0의 첫번째라인을 검정색으로 두번째 라인을 하늘색으로 한다.

● 참　조 : COLOR =, SPRITE, PUT SPRITE, 해설편의 「스프라이트」

COLOR SPRITE (칼라 스프라이트)

스프라이트 면전체의 색을 지정한다.

● 문형식 : COLOR SPRITE (〈면번호〉) = 〈수식〉

● 설　명 : 스프라이트 면전체의 색을 지정한다는 점과, B7의 지정이 불가능하다는 점만 다르고, 나머지는 위의 COLOR SPRITE$과 같다.

COLOR SPRITE(1) = 15 ; 스프라이트 면번호1의 전체를 흰색으로 한다.

CONT (컨티뉴)

[CTRL] + [STOP] 또는 STOP, END 명령에 의해 정지된 프로그램을 다시 실행한다.

● 설　명 : ○ 프로그램의 실행이 정지되었을 때, 이 명령을 실행하면, 정지된 다음 행부터 다시 실행되며, INPUT문에서 실행이 정지된 경우, 정지된 행부터 실행된다.

○ 다음은 CONT 명령으로 프로그램의 실행을 계속할 수 없는 경우이다.

- 실행정지 중에 프로그램의 내용을 변경한 경우
- 프린터에 데이타를 출력하고 있을 때, 정지된 경우
- 데이타 레코더에 출력하고 있을 때, 정지된 경우

● 참　조 : STOP, END

COPY (카피)

화일을 디스켓에 복사한다.

●**문형식** : COPY "[〈드라이브명 1:〉]〈화일명〉[.〈타잎명〉]" TO "[〈드라이브명2:〉]
〈화일명〉[.〈타잎명〉]"

●**설　명** : ○〈화일명〉에 대한 자세한 설명은 해설편의「화일」을 참조하십시오.
○디스크 드라이브가 한대인 경우.
• 같은 디스켓 내에 복사.
〈드라이브명1〉과〈드라이브명2〉는 같거나, 생략할 수 있으며,
〈화일명〉은 다른〈화일명〉으로 지정하여 복사한다. 만약 TO 이하를 생략하면,
〈화일명〉이 같게 되어 에러가 발생한다.
• 다른 디스켓에서의 복사.
두 대의 드라이브가 있다고 생각하고,〈드라이브명1〉과〈드라이브명2〉를
서로 다르게 지정하면,

ⓐ Insert diskette for drive B:
and strike a key when ready.

라는 메세지가 화면에 나타난다. 이 때, 현재 드라이브에 들어있는 디스켓을 꺼내고,
새로운 디스켓(포맷된 것)을 집어넣고, 아무 키나 누른다. 그러면,

ⓑ Insert diskette for drive A:
and strike a key when ready.

라는 두번째 메시지가 나타난다. 새로운 디스켓을 꺼내고 원래의 디스켓을
집어 넣은 다음, 다시 아무 키나 누른다. 그리고 ⓐ의 메시지가 나타나면
새로운 디스켓을, ⓑ의 메시지가 나오면 원래의 디스켓을 집어넣는 작업을
「Ok」가 나타날 때까지 반복한다.「Ok」가 나타나면 끝난 것으로, 이때, 새로운
디스켓에는 지정된〈화일명〉으로 화일이 복사되어 있다.
• 디스크 드라이브가 두대일 때는, 드라이브명을 서로 다르게 지정하면, 램프의 불이
엇갈려 켜지면서 복사된다.
○ * 와 ? 로 여러개의 화일을 한꺼번에 복사할 수 있다. * 는〈화일명〉이나〈타잎명〉에
대응하여 쓰여지고, ? 는 한 문자에 대응하여 쓰여진다.

COPY "ABC. BAS" TO "XYZ. BAS"	; 드라이브명을 모두 생략하였으므로 같은 디스켓에 다른 화일명으로 복사한다.
COPY "A:ABC. BAS" TO "B:"	; 다른 디스켓에 같은 화일명으로 복사한다.
COPY "A:ABC. BAS" TO "B:BCD"	; 다른 디스켓에 다른 화일명으로 복사한다.
COPY "A: * . BAS" TO "B:"	; 타잎명을「BAS」로 하는 모든 화일을 다른 디스켓에 같은 화일명으로 복사한다.
COPY "A:ABC * " TO "B:"	; 화일명이「ABC」로 시작하면서 타잎명이 없는 모든 화일을 다른 디스켓에 같은 화일명으로 복사한다.
COPY "A:PROG ? ? . BAS" TO "B:"	; 화일명이「PROG」와 나머지 어떤 문자로 구성되고 타잎명이「BAS」인 모든 화일을 다른 디스켓에 같은 화일명으로 복사한다.
COPY "A: * . * " TO "B:"	; 디스켓에 있는 모든 화일을 다른 디스켓에 복사한다.

COPY (카피)　　Array to File

배열변수에 있는 내용을 화일에 복사한다.

● 문형식 : COPY 〈배열명〉 TO "〈화일명〉"

● 설　명 : ○ 〈화일명〉에 대한 자세한 설명은 해설편의 「화일」을 참조하십시오.
○ 배열변수에는 비디오 메모리의 내용이 들어있다.

```
10 COLOR 15, 4, 4:SCREEN 5          ;화면색과 화면모드를 설정
20 DEFINT A:DIM A(204)              ;변수 A를 정수형으로 하고 205개의 방을 갖는
                                     배열변수로 한다.
30 LINE (50, 50)-(80, 75), 15, BF   ;흰색의 직사각형을 그리고, 그 안을 흰색으로 칠한다.
40 COPY (50, 50)-(80, 75), 0 TO A   ;페이지0의 (50, 50)부터 (80, 75)까지에 있는 그림을
                                     배열변수A에 복사한다.
50 COPY A TO "A:BOX"                ;배열변수A의 내용을 「BOX」라는 이름으로
                                     디스크 드라이브A에 복사한다.
60 GOTO 60                          ;그래픽화면에서 텍스트화면으로 넘어가지 못하도록 한다.
```

COPY (카피)　　Array or File to Video memory

배열 또는 화일의 내용을 비디오 메모리에 복사한다.

● 문형식 : COPY 〈배열명 또는 "화일명"〉 〔, 〈방향〉〕 TO (〈전송선좌표〉) 〔, 〈전송선페이지〉〕
〔, 〈논리연산기호〉〕

● 설　명 : ○ 〈배열명 또는 화일명〉에 대한 자세한 설명은 해설편의 「화일」을 참조하십시오.
○ 비디오 메모리의 내용이 복사되어 있는 배열이나 화일을 사용하여 다시 비디오 메모리에 복사한다.
○ 〈방향〉의 범위는 0～3이며, 생략하면 0이 된다.
각각의 의미는 다음과 같다.
0 : 좌상에서 우하
1 : 우상에서 좌하
2 : 좌하에서 우상
3 : 우하에서 좌상

```
10 COLOR15, 1, 1:SCREEN 5             ;화면색과 화면모드설정 (SCREEN명령 참조)
20 COPY "A:TEST", 1 TO(100, 100), 0;비디오 메모리의 내용이 복사되어 있는 「TEST」라는
                                     화일의 내용을 화면의 위치(100, 100)에 복사한다.
```

COPY (카피) File to Array

화일에 있는 내용을 배열변수에 복사한다.

● 문형식 : COPY 〈"화일명"〉 TO 〈배열명〉

● 설　명 : ○ 비디오 메모리의 내용이 화일에 저장되어 있을때, 그 내용을 배열변수에 복사한다.
○ 〈화일명〉과 〈배열명〉에 대한 자세한 설명은 해설편의「화일」을 참조하십시오.

```
10 COLOR 15, 4, 4:SCREEN 5;화면색과 화면모드를 설정
20 DIM A%(204)            ;정수형 배열변수 A%의 크기는 205이다.
30 COPY "A:BOX" TO A%  ;디스크 드라이브 A에 있는「BOX」라는 화일의 내용을
                          배열변수 A%에 복사한다.
40 COPY A% TO(100, 100), 0 ;배열변수A%의 내용을 화면의 (100, 100) 위치에 복사한다.
50 END                    ;프로그램의 실행을 끝마친다.
```

COPY (카피) Video memory to Array or File

비디오 메모리의 내용을 배열 또는 화일에 복사한다.

● 문형식 : COPY (〈전송원 좌표1〉)―(〈전송원 좌표2〉) 〔, 〈전송원 페이지〉〕
TO 〈배열명 또는 "화일명"〉

● 설　명 : ○ 〈전송원 좌표1〉부터 〈전송원 좌표2〉까지에 있는 비디오 메모리의 내용을 배열이나 화일에 복사한다. 이 때, 필요한 배열의 크기를 구하는 방법은 다음과 같으며, 배열의 크기는 배열형에 따라 다르다.

정수형 ······(N+1)₩2
단정도형······(N+3)₩4
배정도형······(N+7)₩8

여기서 N은「N=INT(X*Y/K+5)」로 구해진 값이다.
단, X, Y는 가로, 세로의 도트수이며, K는 스크린모드에 따라 다른 정수로
SCREEN 5와 7에서는 2
SCREEN 6에서는 4
SCREEN 8에서는 1값을 갖는다.
즉 예문의 배열크기 259는 다음과 같이 구해진 것이다.
(INT(32*32/2+5)+1)₩2

○ 〈배열명 또는 화일명〉에 대한 자세한 설명은 해설편의「화일」을 참조하십시오.

```
10 COLOR 15, 1, 1:SCREEN 5            ;화면색과 화면모드 설정(SCREEN 명령참조)
20 DEFINT A-Z:DIM A(259)              ;변수를 정수형으로 하고 변수A를 260의 배열방을
                                       갖는 배열변수로 한다.
30 CIRCLE(48, 48), 15, 15, -1, -3.14  ;중심을 (48, 48)로 하고 반지름을 15로 하며,
40 PAINT(40, 40), 8, 15                시작각도, 끝각도를 각각 -1, -3.14로한 부채꼴을,
                                       흰색으로 테두리를 하고 빨간색으로 칠한다.
50 COPY(32, 32)-(63, 63), 0 TO A      ;페이지0의 (32, 32)부터 (63, 63)까지에 있는 그림을
                                       배열변수A에 복사한다.
60 COPY(32, 32)-(63, 63), 0 TO "A:TEST";페이지0의 (32, 32)부터 (63, 63)까지에 있는 그림을
                                       드라이브A에「TEST」라는 화일명으로 복사한다.
70 END
```

● 참　조 : SET PAGE, LINE

COPY (카피)

Video memory to Video memory

비디오 메모리(VRAM)의 내용을 옮긴다.

● 문형식 : COPY (〈전송원좌표1〉) – (〈전송원좌표2〉) 〔, 〈전송원페이지〉〕 TO
(〈전송선좌표〉) 〔, 〈전송선페이지〉〕 〔, 〈논리연산기호〉〕

● 설　명 : ○ 〈전송원좌표1〉부터 〈전송원좌표2〉까지에 있는 비디오 메모리의 내용을 〈전송선좌표〉가 시작되는 곳부터 복사한다. 이 때, 전송원의 영역인 「(전송원좌표1) – (전송원좌표2)」은 LINE문에 의해 지정된다.
○ 〈전송원페이지〉는 전송원, 〈전송선페이지〉는 전송선의 페이지를 지정하며, 생략하면 실행페이지(SET PAGE 명령참조)가 선택된다.
○ 〈논리연산기호〉로는 AND, OR, XOR, PSET, PRESET, TPSET, TPRESET, TXOR, TOR, TAND가 있으며, 만약 생략되면, 「PSET」으로 간주한다.
〈논리연산기호〉에 대한 자세한 설명은 해설편의 「색상부분」을 참조하십시오.

```
10 COLOR 15, 2, 2:SCREEN 5                      ;화면색과 화면모드 설정(SCREEN 명령참조)
20 SET PAGE, 1:CLS:SET PAGE 0, 0                ;페이지1을 깨끗히 지운다.
                                                 (SET PAGE 명령참조)
30 CIRCLE(16, 16), 15, 15:PAINT(16, 16), 15, 15;중점을 (16, 16)으로 하고 반지름을 15로 한
                                                 하얀 원을 그린다.
40 COPY(0, 0)– (31, 31), 0 TO (32, 0), 1, XOR  ;페이지0의 (0, 0)부터 (31, 31)까지에 있는 그림을
                                                 페이지1에 복사한다. 이때, 복사되는 위치는
                                                 (32, 0)을 시작점으로 하고,
                                                 색깔은 화면의 바탕색과 원의 색을 XOR한
                                                 값이다.
50 SET PAGE 1, 1:FOR I =0 TO 99:NEXT           ;페이지1의 내용을 잠깐 화면에 표시한다.
60 SET PAGE 0, 0:FOR I =0 TO 99:NEXT           ;페이지0의 내용을 잠깐 화면에 표시된다.
70 GOTO 50                                      ;실행순서를 50행으로 되돌린다.
```

● 참　조 : SET PAGE, LINE

COPY SCREEN (카피 스크린)

외부 비디오 신호를 디지타이즈하여 비디오 메모리에 써 넣는다.

● 문형식 : COPY SCREEN〔〈모드〉〕

● 설　명 : ○ 외부 비디오 신호를 디지타이즈하여 비디오 메모리로 전송한다.
○ 〈모드〉는 0과 1이다. 〈모드〉가 "0"일 경우에는 한 피일드의 영상을 디지타이즈하고, "1"일 경우에는 두 피일드 즉 홀수페이지와 짝수페이지에 각각 나누어 디지타이즈한다. 이 경우 화면 모드를 인터레이스 모드로 지정해 주어야 한다(초기치는 "0")

> 주의 외부 비디오 신호를 디지타이즈하는 하드웨어(HARD WARE)가 없는 시스템에서는 이 명령을 사용할 수 없다.

COS (코사인)

코사인(COSINE)값을 얻는다.

● 문형식 : COS (〈수식〉)

● 설　명 : ○ 〈수식〉의 단위는 라디안이다.
○ 얻어진 값의 범위는 −1∼+1이며, 배정도 실수형이다.
○ 라디안과 각도의 관계에 대한 설명은 SIN 명령을 참조하십시오.

A=COS(30＊2＊3.14159/360) ; 30×2×3.14159/360 라디안은 30° (도)이므로, A에는 COS 30° 의 값인 0.86602562491683이 대입된다.

● 참　조 : ATN, SIN, TAN

CSAVE (시세이브)

프로그램을 데이타 레코더에 보존한다.

● 문형식 : CSAVE"〈화일명〉" 〔, 〈전송속도〉〕

● 설　명 : ○ 주변장치중 데이타 레코더에서만 사용이 가능하다.
○ 프로그램이 내부어로 보존되기 때문에, MERGE 명령을 사용할 수 없다.
○ 〈화일명〉은 최대 6자까지 가능하며, 초과될 경우는 앞의 6자만 유효하고, 나머지는 무시된다.
○ 〈전송속도〉는 보존할 때의 속도이며, 1(1200보) 또는 2(2400보)이다.
보(BAUD)는 전송속도의 단위로서, 초당 한 비트(Bit)를 전송한다.
〈전송속도〉를 생략하면, SCREEN문에서 지정한 〈전송속도〉로 전송한다.
○ 프로그램 중에 CSAVE 명령을 실행하면, 실행 후 바로 직접 모드로 되돌아가기 때문에, 그 뒤의 프로그램은 실행되지 않는다.
○ CSAVE를 중지시킬 때는 [CTRL] + [STOP]를 누른다.

CSAVE "TEST" ; 프로그램을 「TEST」라는 화일명으로 데이타 레코더에 보존한다.

● 참　조 : CLOAD, CLOAD ? , BSAVE, SAVE, MERGE

CSNG (컨버트 싱글)

단정도 실수형으로 변환한다.

● 문형식 : CSNG (〈수식〉)

● 설　명 : 〈수식〉의 값을 단정도 실수형으로 변환하고, 그 값이 −9.99999E+62∼9.99999E−62의 범위를 벗어나면 "Overflow" (오버플로우) 에러가 발생된다.

A=CSNG(123.45678) ; A에는 123.45678의 단정도 실수값인 123.457이 대입된다.

● 참　조 : CDBL

4 베이직 설명서

CSRLIN (커서라인)

텍스트 화면에서 커서의 수직위치를 조사한다.

●설　명 : ○ 텍스트 화면에서만 사용한다.
○ 현재 커서가 텍스트 화면의 어떤 줄에 있는가를 조사하며, 화면은 맨 윗줄이 1, 마지막 줄이 24이다.
○ 이 시스템 변수에는 값을 대입할 수 없다.

```
Y=CSRLIN    ;화면의 맨 윗줄에서 이 명령을 실행하면, Y에는 1이 대입된다.
```

●참　조 : POS, LOCATE

CVI/CVS/CVD (컨버트 인테져/컨버트 싱글/컨버트 더블)

문자형을 각각 수치형으로 바꾼다.

●문형식 : CVI (〈2바이트 문자〉)
CVS (〈4바이트 문자〉)
CVD (〈8바이트 문자〉)

●설　명 : ○ 수치를 랜덤 디스크 화일에 써 넣을 때는 문자형으로 써 넣어야 한다. 이것을 읽어 내어 다시 수치형으로 바꾸어주는 명령이다.
○ CVI는 2바이트 문자를 정수로, CVS는 4바이트 문자를 단정도 실수로, CVD는 8바이트 문자를 배정도 실수로 변환하는 것이다.

```
10 OPEN "A:NUMBER. DAT" AS #1          ;「NUMBER. DAT」라는 화일을 연다.
20 FIELD #1, 2 AS IN$, 4 AS SG$, 8 AS DB$;하나의 레코더를 변수 IN$에 2자리, SG$에 4자리,
                                          DB$에 8자리로 할당한다.
30 GET #1, 1                           ;화일의 첫번째 레코더를 읽어들인다.
40 PRINT CVI(IN$)                      ;IN$의 값을 정수로 변환하여 화면에 표시한다.
50 PRINT CVS(SG$)                      ;SG$의 값을 단정도 실수로 바꾸어 화면에 표시한다.
60 PRINT CVD(DB$)                      ;DB$의 값을 배정도 실수로 바꾸어 화면에 표시한다.
70 CLOSE #1                            ;열었던 화일을 닫는다.
80 END
```

●참　조 : PUT, MKI$, MKS$, MKD$

DATA (데이타)

READ문의 변수에 할당할 정수를 정의한다.

●문형식 : DATA 〈정해진 수〉 [, 〈정해진 수〉…]

●설　명 : ○ 〈정해진 수〉 즉, 데이타는 READ문의 변수의 형과 일치해야 하며, 한 행의 범위(255문자) 내에서 몇 개라도 쉼표(,)로 구분하여 기입할 수 있다.
○ READ문으로 읽을 데이타보다 DATA문 안의 데이타가 적으면, "Out of DATA" (아웃 오브 데이타) 에러가 일어난다.

```
10 READ A$, B     ;문자형 변수 A$에는 KIM을, 수치형 변수 B에는 56을 각각 대입한다.
20 DATA KIM, 56
```

●참　조 : READ, RESTORE

DEF FN (디파인 펑션네임)

함수식을 정의한다.

● **문형식** : DEF FN〈이름〉〔(〈인수〉〔,〈인수〉…〕)〕=〈함수의 정의식〉

● **설 명** : ○ 〈이름〉은 영문자로 시작하는 2자 이내의 영, 숫자이며, 〈인수〉는 〈함수의 정의식〉 중의 변수와 서로 대응된다.
○ 프로그램 중에 자주 사용하는 연산, 또는 연산에 사용되는 변수를 미리 함수로 정의해 놓고 필요할 때 마다 이 함수를 사용하여 연산 결과를 구할 수 있다.
○ 정의한 함수는 「FN 이름(변수)」의 형식으로 호출한다.
○ 정의할 때의 변수는 임의의 것이므로 호출할 때는 그때그때에 필요한 변수로 바꿔서 지정할 수 있다.

```
10 DEF FN A(X)=X*X*3.14;원의 면적을 구하는 계산식을 A(X)라고 정의한다.
20 FOR  I =10 TO 20 STEP 5  ;변수 I의 값이 10, 15, 20으로 변하면서 NEXT문이 나올때까지의
                              범위 안을 반복 실행한다.
30 PRINT FN A(I)            ; I 값에 대한 원의 면적을 구하여 표시한다.
40 NEXT
```

DEF INT/SNG/DBL/STR (디파인 인테져/싱글/더블/스트링)

변수의 형을 선언한다.

● **문형식** : DEFINT 〈문자의 범위〉〔,〈문자의 범위〉……〕
DEFSNG 〈문자의 범위〉〔,〈문자의 범위〉……〕
DEFDBL 〈문자의 범위〉〔,〈문자의 범위〉……〕
DEFSTR 〈문자의 범위〉〔,〈문자의 범위〉……〕

● **설 명** : ○ 〈문자의 범위〉내에 있는 변수를 다음과 같은 형으로 선언한다.
DEFINT ……정수형
DEFSNG……단정도 실수형
DEFDBL……배정도 실수형
DEFSTR ……문자형
○ 〈문자의 범위〉는 「영문자」 또는 「영문자－영문자」로 지정한다.
○ 이 선언문에서 행한 형선언 보다 형선언 문자(%, !, #)에 의하여 지정된 쪽이 우선이다.
○ 형선언을 하지 않은 문자로 시작하는 변수는 모두 배정도 실수이다.
○ 이 선언문으로 선언한 후에 CLEAR 문을 실행하면, 그 선언이 모두 무효가 된다.

```
DEFINT A, I-K  ;A, I, J, K로 시작되는 변수를 정수형으로 선언한다.
```

DEF USR (디파인 유저)

기계어 프로그램의 실행 번지를 지정한다.

● 문형식 : DEF USR [〈번호〉] = 〈실행번지〉

● 설　명 : ○ USR 함수를 사용하여 호출하는 기계어 프로그램의 〈실행 번지〉를 지정한다.
　　　　○ 〈번호〉는 0～9의 값이고, 기계어 프로그램을 여러개 사용할 경우 구별하는데 쓰인다. 생략되는 경우는 0이 지정된다.

DEF USR2 = &HE000　;　기계어 프로그램2의 실행번지가 &HE000이다.

● 참　조 : CLEAR, USR

DELETE (딜리트)

프로그램의 일부분을 지운다.

● 문형식 : DELETE 〈행번호의 범위〉

● 설　명 : ○ 〈행번호의 범위〉내의 행을 삭제한다.
　　　　○ 〈행번호의 범위〉는 「행번호—행번호」로 나타내며, 행번호는 각각 생략할 수 있으나 동시에 모두 생략할 수는 없다.
　　　　○ 행번호 대신 . (마침표)를 사용하면, 베이직이 현재 가리키고 있는 행번호가 지정된다.

DELETE 50—100; 프로그램의 행번호 50～100을 지운다.
DELETE 10　　　; 행번호 10을 지운다.
DELETE —100　　; 프로그램의 처음부터 행번호 100까지 지운다.
DELETE. —200　; 현재 베이직이 가리키고 있는 행번호부터 행번호 200까지 지운다.

● 참　조 : NEW

DIM (디임)

배열을 정의하고 그것을 메모리 영역에 할당한다.

● 문형식 : DIM 〈변수명〉 (〈첨자의 최대치〉 [, 〈첨자의 최대치〉…])

● 설　명 : ○ DIM은 사용할 배열변수의 최대치를 설정하고, 메모리 안에 그 배열의 영역을 확보한다.
　　　　○ 〈첨자의 최대치〉를 설정할 때, 최소값은 0이고, 최대값은 메모리의 용량이 허용하는 범위이다. 첨자가 10이내인 경우는 DIM문으로 선언하지 않아도 된다.
　　　　○ 선언되어 있는 배열변수는 CLEAR 명령이나 ERASE 명령을 실행하지 않으면 다시 선언할 수 없다.
　　　　○ 필요없는 배열변수는 ERASE 명령으로 지울 수 있다.

```
10 DIM A$(3)      ;A$을 배열변수로 선언하며, 사용할 배열의 최대값은 4로 설정된다.
20 FOR I=0 TO 3   ;변수 I에 0, 1, 2, 3의 값을 차례로 대입하면서 NEXT문까지의 내용을
                   반복한다.
30 READ A$(I)     ;DATA문의 데이타를 차례대로 읽어들인다.
40 PRINT A$(I);   ;읽어들인 데이타를 화면에 표시한다.
50 NEXT
60 DATA M, S, X, 2
```

● 참　조 : CLEAR, ERASE

DRAW(드로우)

그래픽 화면에 도형을 그린다.

● **문형식** : DRAW 〈문자식〉

● **설 명** : ○ 이 명령의 지정 형식은 다음과 같다.

DRAW "문자식"
DRAW 문자변수
DRAW "X 문자변수; "

〈문자식〉은 선을 그리는 정보이며 기호와 수치로 만들어진다.
문자변수에는 미리 〈문자식〉을 대입하여 둔다.
문자변수를 사용할 때는 아래 프로그램에서와 같이 문자변수만(행번호60)으로 지정하거나, "X 문자변수;" (행번호 50)로 지정하여도 결과는 같다.

```
10 SCREEN 2                                        ; 화면모드를 고해상도 그래픽모드로 설정.
20 A$="A0C15S4BM150, 80L40D12R8U6R24D6R8U12"; 회전각도를 0, 칼라를 15로 하고
                                                     시작점을 (150, 80)으로 하여
                                                     그림을 그리도록 정의한 것이다.
30 B$="BM134, 86F12D8L32U8E12"                     ; 시작점을 (134, 86)으로 하여
                                                     그림을 그리도록 정의한 것이다.
40 C$="C8BM132, 92L4G4D4F4R4E4U4H4"                ; 칼라를 8로 하고 시작점을
                                                     (132, 92)로 하여 그림을 그리도록
                                                     정의한 것이다.
50 DRAW "XA$;":DRAW "XB$;"                          ; 위에서 정의한 것으로
                                                     그림(전화기)을 그린다.
60 DRAW C$
70 GOTO 70                                          ; 그래픽모드에서 정지해 있도록 한다.
```

○ 지정하는 기호에는 이동, 회전, 색상, 이동단위의 4종류가 있다.

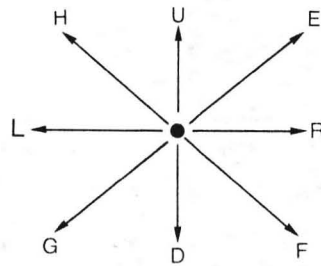
*이동명령

U〈거리〉: 위로 이동
D〈거리〉: 아래로 이동
L〈거리〉: 왼쪽으로 이동
R〈거리〉: 오른쪽으로 이동
E〈거리〉: 오른쪽 위로 이동
F〈거리〉: 오른쪽 아래로 이동
G〈거리〉: 왼쪽 아래로 이동
H〈거리〉: 왼쪽 위로 이동

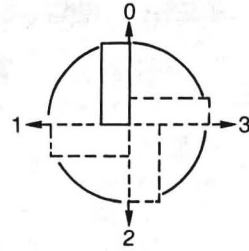
〈거리〉는 화면의 한점을 단위로 한다(단위는 「이동단위」명령으로 바꿀 수 있다).

M x, y : 이동위치를 각각 x방향, y방향에 대하여 절대좌표 또는 상대좌표로 지정한다.
즉 x의 앞에 「+」 또는 「−」를 붙이면, 상대좌표로 간주한다.

B : 각각의 방향으로 이동하지만 그림은 그리지 않으며, U부터 M까지의 기호 앞에 붙여서 사용한다.

N : 각각의 방향으로 그림을 그리면서 이동하고 끝나면 처음 시작점 좌표로 되돌아온다.
U부터 M까지의 기호 앞에 붙여서 사용한다.

＊회전명령

A <각도>: 이동명령으로 그린 그림을 <각도>만큼 회전시킨다.
<각도>는 한번 지정하면 다음에 지정할 때까지 유효하며 0～3이다.

0:0°　　2:180°
1:90°　　3:270°

＊색상명령

C <색번호>: 지정한 색으로 그림을 그린다.
<색번호>는 각 SCREEN 모드에 따라 다르며, 이것에 대해서는 COLOR 명령을 참조하십시오.

＊이동단위 명령

S <정수>: 이동거리의 단위를 설정한다.
<정수>는 0～255이며 이 값의 1/4이 단위거리로 설정된다. 그러므로 이 단위거리와 이동명령으로 지정한 거리를 곱한 것이 화면상에서의 실제 이동거리로 된다.
<정수>를 지정하지 않으면 값은 4이다.

＊각 명령들의 파라메터 즉 <거리>나 <각도> 등을 다음과 같이 수치변수로 지정할 수 있다.
DRAW "＝변수명;"
이 경우 변수명 뒤에는 세미콜론(;)을 반드시 넣어야 한다.

```
5 SCREEN 2
10 X1=40:X2=50                   ;X1과 X2에 x와 y좌표를 각각 대입한다.
20 DRAW "BM=X1;, =X2;U20"; 화면의 (40, 50)을 시점으로 하여 위로 20만큼
                                 이동하며 그린다.
```

DSKF (디스크프리)

디스켓의 남은 용량을 클러스터(Cluster)단위로 표시한다.

● **문형식**: DSKF (드라이브 번호)

● **설　명**: (드라이브 번호)는 현재 사용중인 드라이브가 지정될 때 0, A 드라이브는 1, B 드라이브는 2로 지정한다.
드라이브를 2개 이상 연결한 경우에는 같은 방법으로 C 드라이브는 3, D 드라이브는 4로 지정한다.

DSKF(0)　;현재 사용중인 드라이브의 디스켓에 남은 용량이 클러스터 단위로 표시된다.

참고 MSX시스템에서는 1CLUSTER＝2섹터이다. (1섹터＝512바이트)

DSKI$ (디스크 인 스트링)

섹터의 내용을 읽어서 메모리에 옮긴다.

● 문형식 : DSKI$ (〈드라이브 번호〉, 〈논리섹터 번호〉)

● 설 명 : ○ 〈논리섹터 번호〉로써 지정된 섹터의 내용을 읽어내어, 포인트(F351H, F352H)가 가리키고 있는 메모리 번지부터 써 넣는다.
○〈드라이브 번호〉는 0일때 현재 사용중인 디스크 드라이브, 1일때 드라이브A, 2일때 드라이브B 등으로 각각 대응된다.
○ 〈논리섹터 번호〉는 0부터 시작한다. 그러나 지정된 〈논리섹터 번호〉가 유효한가를 체크하지는 않는다.
○ 섹터의 내용을 메모리에 옮겼다 하더라도 디스크에 관계된 명령 (예를들면, FILES, OPEN, CLOSE···)을 실행하게 되면, 그 메모리의 내용이 파괴되므로 주의하십시오.

```
10 POKE &HF351, &H00;메모리 번지의 하위 바이트를 지정한다.
20 POKE &HF352, &H90;메모리 번지의 상위 바이트를 지정한다.
30 PRINT DSKI$(0, 0)  ;현재 사용하고 있는 디스크 드라이브의 논리섹터 0번째의 내용을 지정한
                       번지로 읽어들인다.
```

● 참 조 : DSKO$

DSKO$ (디스크 아웃 스트링)

메모리의 내용을 섹터에 보존한다.

● 문형식 : DSKO$〈드라이브 번호〉, 〈논리섹터 번호〉

● 설 명 : ○ 포인터(F351H, F352H)가 가리키고 있는 메모리 번지의 내용을 〈논리섹터 번호〉로써 지정된 섹터에 써 넣는다.
○ DSKI$명령과 역할은 다르지만, 사용방법이 같으므로 그것을 참조하십시오.

```
10 POKE &HF351, &H00;메모리 번지의 하위 바이트를 지정한다.
20 POKE &HF352, &H90;메모리 번지의 상위 바이트를 지정한다.
30 DSKO$ 0,15        ;지정한 번지의 내용을 현재 사용하고 있는 드라이브의
                      논리섹터 15번째에 써넣는다.
```

END (엔드)

프로그램의 실행을 끝낸다.

● 설 명 : ○ 프로그램의 실행을 끝내고, 모든 화일을 닫는다.
○ 프로그램 마지막의 END문은 생략할 수 있으며, 이 경우 열려있던 화일은 닫히지 않는다.

```
10 PRINT "ABC";문자열「ABC」를 화면에 표시한다.
20 END         ;프로그램의 실행을 끝낸다.
30 PRINT "DEF";윗 행에서 END명령을 하였으므로, 문자열「DEF」는 화면에 표시되지 않는다.
```

EOF (엔드오브화일)

화일이 끝났는가를 조사한다.

● 문형식 : EOF (〈화일번호〉)

● 설 명 : ○ 〈화일번호〉로 지정된 화일이 끝났는지 조사하며, 끝났으면 −1, 끝나지 않았으면 0을 얻는다.
○ 〈화일번호〉는 OPEN 명령으로 지정한 것으로 INPUT 모드로 열려있어야 한다.

```
10 OPEN "CAS:SAMPLE" FOR INPUT AS #1; 카세트테이프에 있는 「SAMPLE」이라는 화일을
                                      연다.
20 INPUT #1, A$                      ; 화일에서 하나의 레코드를 읽어낸다.
30 PRINT A$                          ; 읽어낸 정보를 화면에 표시한다.
40 IF EOF(1) THEN CLOSE #1:END ELSE 20 ; 화일이 종료하였는지를 조사하여
                                      종료하였을 경우는 화일을 닫고
                                      프로그램의 실행을 끝내고, 아니면 행번호
                                      20으로 점프한다.
```

● 참 조 : OPEN, INPUT #

ERASE (이레이스)

배열변수를 지운다.

● 문형식 : ERASE 〈배열변수명〉 〔, 〈배열변수명〉…〕

● 설 명 : ○ DIM으로 지정한 배열변수를 지우고, 할당되어 있던 메모리를 사용할 수 있게 한다.
○ 이 명령으로 지워진 배열변수는, DIM 명령으로 다시 지정할 수 있다.

```
ERASE A$ ; DIM 명령을 사용하여 배열변수로 지정된 A$을 지운다.
```

● 참 조 : DIM

ERL/ERR (에러라인/에러)

에러가 발생한 행번호나 에러코드를 얻는다.

● 설 명 : ○ 에러가 발생했을 때 ERL은 에러가 발생한 행번호를, ERR은 에러코드값을 갖는다.
○ ERL과 ERR은 ON ERROR GOTO문에 의한 에러처리루틴에서 어떤 에러가 어느 곳에서 일어났는가를 판별할 때 사용한다.
○ 직접(DIRECT) 모드에서 에러가 발생한 경우, ERL 값은 65535로 된다.
○ 에러코드에 대해서는 자료편의 「에러코드」항을 참조하십시오.

```
A=ERL ; A에는 에러가 발생한 행번호가 대입된다.
B=ERR ; B에는 발생한 에러의 코드가 대입된다.
```

● 참 조 : ON ERROR GOTO

ERROR (에러)

프로그램을 에러 상태로 만든다.

● 문형식 : ERROR 〈에러코드〉

● 설　명 : ○ 〈에러코드〉는 0～255의 범위이며, 〈에러코드〉에 대한 의미는 자료편의「에러코드」항을 참조하십시오.

○ 메시지가 정의되어 있는 〈에러코드〉를 지정하면, 그 코드에 대응하는 에러 메시지가 화면에 표시되고, 명령대기 상태로 돌아간다.

○ 사용자 자신의 에러처리를 하고 싶은 경우, 베이직에서 지정되어 있지 않은 〈에러코드〉를 지정하고, ON ERROR GOTO문의 에러처리 루틴에 의해서 에러를 처리할 수 있다.

ERROR 4 ; "Out of DATA" (아웃오브 데이타) 라는 에러 메시지가 나타난다.

● 참　조 : ON ERROR GOTO

EXP (엑스포넨셜)

「e」의 멱승을 구한다.

● 문형식 : EXP (〈수식〉)

● 설　명 : ○ 자연대수의 밑수 e의 〈수식〉승을 값으로 구한다.

○ 〈수식〉의 값은 145.06286085862까지이며, 이 보다 큰 값이면 "Overflow" (오버플로우) 에러가 된다.

○ 결과는 배정도형 실수이다.

A＝EXP(2) ;A에는 7.38905609893이 대입된다.

FILED (필드)

랜덤화일 버퍼에 변수의 길이를 할당한다.

● 문형식 : FIELD〔#〕〈화일버퍼번호〉, 〈문자길이〉 AS 〈문자변수〉〔, 〈문자길이〉 AS 〈문자변수〉…〕

● 설　명 : ○ 〈화일버퍼번호〉는, OPEN문으로 열은 화일버퍼에 붙인 번호이며, 〈문자길이〉는, 〈문자변수〉에 할당되는 문자수이다. 또, 〈문자변수〉는 랜덤화일을 액세스(access) 할 때에 사용되는 변수명이다.

○ FIELD문으로 할당되는 총 바이트 수는 OPEN문에서 결정한 레코드 길이이고, OPEN문에서 결정하지 않았을 경우는 255바이트로 정해진다. 이것을 초과하면, "FIELD overflow" (필드 오버플로우) 에러가 발생한다.

○ 한 개의 〈화일 버퍼번호〉에 대해서 FIELD문을 여러개로 나누어서 실행해도 상관없다. 이 때, 이전에 실행된 FIELD문도 그 효력을 가진다.

○ FIELD문은 실행할 때마다 최초의 한 문자부터 버퍼(buffer)를 정의하기 때문에 같은 데이타에 대해서 여러개의 영역을 정의하는 효과를 갖는다.

○ FIELD문은 GET문에 의하여 랜덤화일 버퍼로부터 데이타를 읽거나, PUT문에 의하여 랜덤화일 버퍼로 데이타를 쓰는 일을 가능하게 하기 위해서 GET, PUT문 전에 실행되어야 한다.

○ FIELD문에서 정의된 〈문자변수〉는 다른 곳에서 사용할 수 없다.

FIELD #1, 15 AS A$, 10 AS B$; 랜덤화일 버퍼에 최초의 15문자(byte)를 A$라고 하는 문자변수에 할당하고, 다음의 10문자를 B$에 할당한다.

● 참　조 : OPEN

FILES/LFILES (화일즈/라인화일즈)

디스켓(diskette)속의 화일명을 화면 또는 프린터에 출력한다.

● 문형식 : 〔L〕 FILES〔"〔〈장치명 : 〉〕 〈화일명〉 〔. 〈타잎명〉〕"〕

● 설　명 : ○ 〈화일명〉, 〈타잎명〉에 대한 설명은 해설편의 「화일」을 참조하여 주십시오.

○ 〈장치명 : 〉으로 지정된 드라이브에 있는 디스켓의 〈화일명〉을 표시하며, 〈장치명 : 〉이 생략되면, 현재 사용중인 드라이브가 지정된다.

○ 〈장치명 : 〉과 〈화일명〉을 생략하면, 현재 사용중인 드라이브에 있는 모든 화일을 표시한다.

○ 〈화일명〉과 〈타잎명〉대신에 *(asterisk), 그 중의 한 문자대신에 ? (의문부호)를 사용할 수 있다. 이것에 대한 설명은 아래의 예를 참조하여 주십시오.

○ 지정된 〈화일명〉이 디스켓에 없으면, "File not found" (화일 낱 파운드) 에러가 발생한다.

○ LFILES는 화일을 프린터(printer)에 출력하라는 뜻이다.

FILES "B:MSX. BAS" ; 화일명이 「MSX. BAS」라는 화일을 드라이브B에서 읽어내어 화면에 표시한다.
FILES "MSX" ; 현재 사용중인 드라이브로부터 「MSX」라는 화일을 읽어내어 화면에 표시한다.
FILES "B:" ; 드라이브B에 있는 모든 화일을 읽어내어 화면에 표시한다.

FILES ; 현재 사용중인 드라이브에 있는 모든 화일을 읽어내어 화면에 표시한다.
FILES "*. *" ; FILES 명령과 같은 결과를 얻는다.
FILES "*. BAS" ; 타잎명을 「BAS」로 하는 모든 화일을 현재 사용중인 드라이브에서 읽어내어 화면에 표시한다.
FILES "A:MSX. *" ; 화일명이 MSX인 모든 화일을 드라이브A에서 읽어내어 화면에 표시한다.
FILES "A:MS ?" ; 화일명이 MS로 시작되는 3문자의 화일들을 화면에 표시한다.
FILES "B:MSX. BA?"; 화일명이 MSX이며 타잎명의 앞쪽에 BA라는 글자가 들어가는 모든 화일을 드라이브B에서 읽어내어 화면에 표시한다.

FIX (픽스)

정수값을 얻는다.

● 문형식 : FIX (〈수식〉)

● 설　명 : 〈수식〉의 소숫점 이하를 버리고, 정수값을 얻는다.

A=FIX(−1.2) ; A에는 −1.2의 소숫점 이하를 버린 −1이 대입된다.

● 참　조 : INT

FOR~NEXT (포오~넥스트)

FOR부터 NEXT까지를 지정된 횟수만큼 반복한다.

●문형식 : FOR〈변수명〉=〈초기값〉 TO 〈최종값〉 〔STEP〈증감〉〕
∫
NEXT〔〔〈변수명〉〕〔,〈변수명〉〕…〕

●설 명 : ○ 〈변수명〉은 반복횟수의 카운터역할을 한다. 이 카운터는, FOR~NEXT문이 1회 실행될 때마다 〈증감〉만큼 증가 또는 감소하며, 그 값이 〈최종값〉을 넘으면, NEXT 다음행으로 이동한다.
○ 〈초기값〉, 〈최종값〉, 〈증감〉은 변수 또는 식이며, STEP이 생략되면, 〈증감〉은 1로 된다.
○ 〈증감〉은 음수값을 취할 수도 있다. 단, 이 경우 〈초기값〉이 〈최종값〉보다 커야 한다.
○ NEXT 문의 〈변수명〉을 생략하면, 제일 가까운 FOR 문과 대응된다.

```
10 FOR I=0 TO 10 STEP 2; 변수를 I, 초기값을 0, 최종값을 10, 증감을 2로 설정한다.
20 PRINT I              ;I의 값을 화면에 표시한다.
30 NEXT I               ;FOR문의 마지막을 지정한다.
```

즉 변수I는 0, 2, 4, 6, 8, 10으로 증가되고 FOR~NEXT문 사이의 PRINT명령은 6번 실행된다.

FRE (프리)

현재 사용할 수 있는 메모리의 크기를 조사한다.

●문형식 : FRE (〈인수〉)

●설 명 : ○ 〈인수〉가 수치일 경우는, 베이직이 사용하지 않은 메모리의 크기를 바이트수로 구한다.
○ 〈인수〉가 문자일 경우는 아직 사용하지 않은 문자열 영역을 바이트 수로 얻으며 문자열 영역중의 쓰지 않는 문자열을 정리함으로써 사용 가능한 문자열 영역을 넓히는 효과를 얻는다.

```
A=FRE(0)   ;A에는 베이직이 사용하지 않은 메모리의 크기가 바이트 수로 대입된다.
B=FRE(A)   ;B에는 아직 사용하지 않은 문자영역의 크기가 바이트 수로 대입된다.
```

GET (겟)

데이타를 랜덤화일버퍼로부터 읽어들인다.

●문형식 : GET〔#〕〈화일번호〉〔,〈레코드번호〉〕

●설 명 : ○ 〈화일번호〉는 OPEN 명령으로 지정한 것과 같아야 한다.
○ 〈레코드번호〉는 최대 4294967295까지 지정할 수 있으며, 이것을 생략하면, 바로 전에 GET문을 실행한 레코드 다음의 레코드가 랜덤화일버퍼에서 읽어진다.

```
10 OPEN "SAMPLE. DAT" AS #1; 「SAMPLE. DAT」라는 화일을 연다.
20 FILED #1, 2 AS A$, 10 AS B$  ; A$은 2자리, B$은 10자리로 지정한다.
30 FOR I%=1 TO 10               ;I%의 초기값은 1, 최종값은 10, 증분은 1로 정한다.
40 GET #1, I%                   ;I%가 갖은 값을 번호로한 레코드를 읽어들인다.
50 PRINT CVI(A$);B$             ; 읽어들인 데이타를 화면에 표시하는데, A$의 내용은
                                 수치형으로 바꾼다.
60 NEXT                         ;FOR문의 마지막을 뜻한다.
70 CLOSE #1                     ;화일을 닫는다.
80 END
```

GET DATE (겟데이트)

현재의 날짜를 얻는다.

● 문형식 : GET DATE 〈문자변수〉 〔,A〕

● 설　명 : ○ 현재의 날짜(년/월/일)를 읽어, 〈문자변수〉에 대입한다.
　　　　○ ,A를 지정하면, 설정된 알람(ALARM)의 날짜를 읽어 〈문자변수〉에 대입한다.

GET DATE A$;A$에 현재의 날짜가 대입된다.
GET DATE A$, A ;A$에 알람으로 기억된 날짜가 대입된다.

● 참　조 : SET DATE, GET TIME

GET TIME (겟타임)

현재의 시각을 얻는다.

● 문형식 : GET TIME 〈문자변수〉 〔,A〕

● 설　명 : ○ 현재의 시각(시 : 분 : 초)을 읽어, 〈문자변수〉에 대입한다.
　　　　○ ,A를 지정하면, SET TIME에서 알람으로 기억된 시각을 읽어 〈문자변수〉에 대입한다.

GET TIME A$;A$에 현재의 시각이 대입된다.
GET TIME A$, A ;A$에 알람으로 기억된 시각이 대입된다.

● 참　조 : SET TIME, GET DATE

GOSUB (고서브)

지정한 서브루틴(SUBROUTINE)을 호출한다.

● 문형식 : GOSUB 〈행번호〉

● 설　명 : ○ 〈행번호〉부터 시작하는, 서브루틴을 부른다.
　　　　○ 서브루틴에서 되돌아오기 위해서는 RETURN 문을 사용한다.
　　　　○ 서브루틴이라는 것은 독립된 한 개의 프로그램으로, GOSUB 문에서 지정한 〈행번호〉부터, RETURN 까지의 명령을 말한다.
　　　　○ 한개의 서브루틴 안에서 다른 서브루틴을 호출할 수도 있다. 이것은 메모리의 스택영역의 용량이 허용하는 한 가능하며, 스택영역이 부족한 경우에는 "Out of memory" (아웃 오브 메모리)가 발생한다.
　　　　○ 서브루틴내에 CLEAR 문이 있어서는 안된다.

GOSUB 30 ;행번호 30부터 시작되는 서브루틴을 부른다.

● 참　조 : RETURN, GOTO

GOTO (고투)

지정된 행으로 점프한다.

●**문형식** : GOTO 〈행번호〉

●**설 명** : ○ 〈행번호〉에서 지정하는 행으로 점프한다.
○ 프로그램에 없는 〈행번호〉를 지정하면 "Undefined line number" (언디파인라인 넘버)라는 에러 메시지가 나온다.

```
10 PRINT "10행 입니다":GOTO 30;문자열「10행 입니다」를 표시하고 행번호 30으로 점프한다.
20 PRINT "20행 입니다":END    ;문자열「20행 입니다」를 표시하고 프로그램을 끝낸다.
30 PRINT "30행 입니다"         ;문자열「30행 입니다」를 표시한다.
40 GOTO 20                    ;행번호 20으로 점프한다.
```

●**참 조** : GOSUB, ON GOTO/ON GOSUB

HEX$ (헥스 스트링)

수식을 16진수 표기의 문자열로 변환한다.

●**문형식** : HEX$(〈수식〉)

●**설 명** : ○ 〈수식〉의 범위는 −32768～65535이다.
○ 결과는 0～9의 숫자와 A～F의 문자로 구성되며 최대 4자리까지 나타낼 수 있으나 맨앞부터 계속되는 0은 표시되지 않는다.

```
A$=HEX$(12)  ;십진수 12에 대응하는 16진수의 값 C가 변수 A$에 대입된다.
```

●**참 조** : VAL, BIN$, OCT$, STR$

IF～THEN～ELSE/IF～GOTO～ELSE (이프～덴～엘스/이프～고투～엘스)

조건을 판단하여, 그 결과에 따라 프로그램의 흐름을 결정한다.

●**문형식** : IF〈조건〉 THEN 〈문 또는 행번호〉 〔ELSE〈문 또는 행번호〉〕
IF〈조건〉 GOTO 〈행번호〉 〔ELSE〈문 또는 행번호〉〕

●**설 명** : ○ 〈조건〉을 만족하면 THEN 뒤에 지정된 것을 실행하고, 그렇지 않을 때는 ELSE 뒤에 지정한 것을 실행한다.
○ THEN 또는 ELSE 다음에 〈행번호〉대신 〈문〉이 지정되면, 해당 〈문〉을 실행한 후 바로 다음 행으로 간다.
○ THEN 뒤에 〈행번호〉를 지정할 경우,「IF～GOTO～ELSE」라는 형식을 사용할 수 있다.
○ 〈조건〉에는 다음과 같이 비교연산기호와 논리연산기호가 사용된다. 논리연산기호는 여러개의 조건을 지정할 때 사용한다.

```
10 A=A+1                          ;변수 A의 값에 1을 더한다.
20 PRINT A                        ;변수 A의 값을 표시한다.
30 IF A=20 THEN PRINT "END" ELSE 10;변수 A의 값이 20이면「END」를 표시하고
                                   아니면 10행으로 점프한다.
40 END
```


4 베이직 설명서

비교연산 기호	의 미	예
식1=식2	식1이 식2와 같다.	IF A$="X"
식1<>식2	식1이 식2와 같지않다.	IF A<>0
식1<식2	식1이 식2보다 작다.	IF A<10
식1>식2	식1이 식2보다 크다.	IF A>B
식1<=식2 (식1=<식2)	식1이 식2보다 작거나 같다.	IF A<=B+1
식1>=식2 (식1=>식2)	식1이 식2보다 크거나 같다.	IF A$>="Z"

논리연산 기호	예	의 미
NOT	IF NOT(A=1)	A가 1이 아니면 THEN 뒤를 실행
AND	IF A=0 AND B=3	A가 0이고 B가 3이면 THEN 뒤를 실행
OR	IF A<0 OR B=0	A가 0보다 작거나 B가 0이면 THEN 뒤를 실행

INKEY$ (인키스트링)

키 입력을 받아들인다.

● 설 명 : ○ 키를 누르면 그 값을 갖고, 누르지 않으면 널스트링을 얻는다.
○ [CTRL]+[C], [CTRL]+[STOP] 은 INKEY$ 함수에 의하여 얻을 수 없다.
○ 입력된 키는 표시되지 않는다.

```
10 A$=INKEY$            ;키를 누르면 눌러진 문자를, 누르지 않으면 널스트링을 A$에 대입한다.
20 IF A$=" " THEN 10;키를 받지 않았으면 행번호 10으로 점프한다.
30 PRINT A$             ;A$에 대입된 문자를 표시한다.
40 GOTO 10              ;행번호 10으로 가서 계속 키를 받게한다.
```

● 참 조 : INPUT, LINE INPUT, INPUT$, 자료편의 「문자코드표」

INP (인풋포트)

입력 포트(PORT)로 부터 값을 얻는다.

● 문형식 : INP (〈포트번호〉)

● 설 명 : 〈포트번호〉에서 지정한 입력 포트로 부터 1바이트(8비트)를 읽어 그것을 함수값으로 한다.
지정 가능한 값은 0부터 255까지이다.

```
A=INP (15)   ;포트번호 15로부터 데이타를 읽어들여, 그 값을 변수A에 대입한다.
```

● 참 조 : OUT, WAIT, 자료편의 「I/O맵」

INPUT (인풋)

입력된 데이타를 지정한 변수에 대입한다.

●문형식 : INPUT〔"〈프롬프트문〉" |;|〕〈변수명〉〔,〈변수명〉……〕

●설 명 : ○ 이 명령을 실행하면, 의문부호(?)를 화면에 표시하고, 프로그램은 키보드로부터 데이타의 입력을 기다린다.

○ 〈프롬프트문〉을 지정하면 의문부호 앞에 그 문자열이 표시된다.

○ 변수를 쉼표(,)로 구분하여 여러개 지정하는 경우에는, 입력하는 데이타도 쉼표로 구분하여 변수의 갯수만큼 입력한다. 입력한 데이타의 갯수가 부족하면, "? ?"를 표시하고 입력을 기다리며, 갯수가 많으면 "?Extra ignored"를 표시하고 남은 데이타는 무시된다. 대응하는 변수와 데이타의 형은 일치해야 한다. 대응하는 형이 다를 경우는, "Redo from start"를 표시하고 다시 입력을 기다린다.

```
INPUT A    ;「?」를 표시하고 키보드로 부터 수치가 입력되면 A에 대입한다.
```

INPUT # (인풋 넘버)

화일로 부터 데이타를 읽어 변수에 대입한다.

●문형식 : INPUT #〈화일번호〉,〈변수〉〔,〈변수〉……〕

●설 명 : ○ 〈화일번호〉는 OPEN에서 지정한 번호이고, INPUT모드로 지정한 화일로 부터 데이타를 읽어 변수에 대입한다.

○ INPUT #문에서 읽어 들이는 데이타는 PRINT #문에 의해서, 지정되어 있는 데이타이다.

○ 〈변수〉의 형은 입력되는 데이타의 형과 같아야 한다.

○ 읽어들인 데이타의 첫 문자가 공백이거나, 캐리지리턴 또는 라인피드가 아니면 문자열 또는 수치의 시작으로 본다.

○ 수치 데이타는 공백이나 캐리지리턴, 라인피드 또는 쉼표로 구분된다.

○ 문자열은 쉼표나 캐리지리턴, 라인피드로 구분되며, 최대 255문자까지 읽을 수 있다.

○ 처음 문자가 인용부호(")이면 다음에 인용부호가 나올때 까지의 모든 문자를 문자열로 간주한다. 그러므로 문자열 안에는 인용부호가 들어갈 수 없다.

```
10 OPEN "CAS:SAMPLE" FOR INPUT AS #1;데이타 레코더에 있는「SAMPLE」이라는
                                     화일을 연다.
20 INPUT #1, A$                     ;열려진 화일에서 데이타를 읽어 낸다.
30 PRINT A$                         ;읽어낸 데이타를 화면에 표시한다.
40 END
```

●참 조 : INPUT, OPEN, PRINT #

INPUT$ (인풋 스트링)

화일로 부터 문자열을 읽어낸다.

● **문형식** : INPUT$(〈식〉〔,〔#〕〈화일번호〉〕)

● **설　명** : ○ OPEN문에서 INPUT모드로 지정한 화일로 부터 〈식〉만큼의 데이타를 읽어, 하나의 문자열로 한다.
○ 〈식〉은 읽어낼 문자의 길이를 지정한다.
○ 지정한 〈식〉 만큼의 문자가 입력될 때까지 기다리며, 이미 버퍼에 입력된 데이타가 있을 경우 버퍼속에 있는 문자를 함께 얻게 된다.
○ 〈화일번호〉를 생략하면, 키보드로 부터 입력이 되며, 입력된 문자는 화면에 표시되지 않는다.
○ 이 명령은 [CTRL] + [STOP] , [CTRL] + [C]를 제외한 모든 문자를 읽어내므로, INPUT문이나 LINE INPUT문으로 입력할 수 없는 라인 피드 코드 또는 캐리지리턴 코드를 입력할 수 있다.

A$=INPUT$(10, #2) ; 화일번호 2를 열어놓은 화일에서 10문자 만큼의 문자열을 읽어내어 A$에 대입한다.
B$=INPUT$(5) ; 키보드로 부터 5문자 만큼의 문자열을 읽어들여 B$에 대입한다.

● **참　조** : OPEN, INPUT, LINE INPUT

INSTR (인스트링)

문자열 안에서 지정하는 문자열을 찾아, 그 문자의 위치를 얻는다.

● **문형식** : INSTR(〔〈수식〉,〕〈문자열1〉, 〈문자열2〉)

● **설　명** : ○ 〈문자열1〉안에서 〈문자열2〉를 찾아, 그 위치를 얻는다.
○ 〈수식〉은 조사할 시작 위치를 0～255의 범위로 지정한다.
이 〈수식〉을 생략할 경우, 〈문자열1〉의 처음부터 조사하기 시작한다.
○ 다음과 같은 경우에는 결과가 0이 된다.
- 문자열 안에 찾고 있는 문자열이 없을 때.
- 〈문자열2〉의 길이가 〈문자열1〉보다 클 때.
- 〈문자열1〉이 널 스트링인 경우.

A$="SUPER MSX" 일때
Y=INSTR(6, A$, "M") ; A$에 대입된 문자열을 여섯번째 문자부터 순서대로 조사하여 "M"이 찾아지면, 그 위치인 7이 Y에 대입된다.
X=INSTR(A$, "G") ; A$에 대입된 문자열 안에 "G"가 없으므로, X에는 0이 대입된다.

● **참　조** : LEN

INT (인테져)

값을 넘지않는 최대정수를 구한다.

●문형식 : INT (〈수식〉)

●설　명 : 〈수식〉의 값을 넘지않는 최대정수 값을 구한다.

A=INT(−1.2)　;A에는 −2를 넘지않는 최대정수가 대입된다.

●참　조 : FIX, CINT

INTERVAL ON/OFF/STOP (인터벌 온/오프/스톱)

ON INTERVAL GOSUB 명령의 실행을 결정한다.

●문형식 : INTERVAL ON
INTERVAL OFF
INTERVAL STOP

●설　명 : ○ ON INTERVAL GOSUB문에서 설정한 타임 인터럽트를 허가(ON), 금지(OFF) 또는 보류(STOP)한다.
○ INTERVAL ON 명령을 실행하면, ON INTERVAL GOSUB문에서 지정해준 시간마다 인터럽트가 걸려 실행 중인 프로그램을 중지하고 인터럽트 처리루틴을 실행한다.
○ INTERVAL OFF 명령을 실행하면, ON INTERVAL GOSUB문에서 지정한 시간이 지나도 인터럽트가 일어나지 않는다.
○ INTERVAL STOP을 실행하면 타임 인터럽트가 체크되더라도 INTERVAL ON이 실행될 때까지 서브루틴 처리가 보류된다. INTERVAL ON을 실행하면, 즉시 타임 인터럽트 처리 루틴을 허가한다.

●참　조 : ON INTERVAL GOSUB

KEY (키)

기능키 (Function Key) 의 내용을 정의한다.

●문형식 : KEY〈키번호〉, 〈문자열〉

●설　명 : ○ 〈키번호〉에 해당하는 기능키의 내용을 〈문자열〉로 정의한다.
○ 〈키번호〉는 1~10이고, 각각 키보드의 [F1] ~ [F10] 에 대응한다.
○ 〈문자열〉은 최대 15문자까지의 문자 및 제어코드로 정의된다.
○ 정의된 기능키의 내용은 다시 정의하든지, 또는 시스템이 리세트될 때까지 변하지 않는다.
○ CHR$ 함수를 +로 연결하여 사용할 수 있다.

KEY1, "LOAD"　;[F1] 의 내용을 「LOAD」라고 정의한다.
KEY2, "RUN"+CHR$(13) ;[F2] 의 내용을 「RUN」 + [RETURN] 으로 정의한다.

●참　조 : CHR$, KEY LIST, KEY ON/OFF

KEY LIST (키리스트)

기능키의 내용을 화면에 표시한다.

● 설 명 : ○ KEY 명령에 의해 정의된 기능키의 내용이 잘 되어있는지를 확인한다.
○ [F1] ~ [F10] 의 기능키 내용을 화면에 표시한다.
○ 처음에는 다음과 같이 설정되어 있다.

[F1] ———— "color"
[F2] ———— "auto"
[F3] ———— "goto"
[F4] ———— "list"
[F5] ———— "run"+CHR$(13)
[F6] ———— "color 15, 4, 7"+CHR$(13)
[F7] ———— "cload"+CHR$(34)
[F8] ———— "cont"+CHR$(13)
[F9] ———— "list."+CHR$(13)+CHR$(30)+CHR$(30)
[F10] ———— CHR$(12)+"run"+CHR$(13)

● 참 조 : KEY; KEY ON/OFF

KEY(n)ON/OFF/STOP (키온/오프/스톱)

기능키를 눌렀을 때, 지정된 프로그램 처리를 행하는가를 결정한다.

● 문형식 : KEY (〈키번호〉) ON
KEY (〈키번호〉) OFF
KEY (〈키번호〉) STOP

● 설 명 : ○ ON KEY GOSUB 문에서 지정한 서브루틴의 실행을 허가(ON), 금지(OFF), 보류(STOP) 하는 것이다.
○ 〈키번호〉는 [F1] 에서 [F10] 까지의 기능키를 표시한다.
○ KEY(n) ON 명령을 실행하면 지정한 기능키를 누를때에 ON KEY GOSUB 문에서 정의되어 있는 서브루틴을 실행한다.
○ KEY(n) OFF를 실행하면 지정한 기능키를 눌러도 서브루틴을 실행하지 않는다.
○ KEY(n) STOP을 실행하면 기능키를 눌러도 다시 KEY(n) ON을 실행할 때까지 서브루틴의 실행을 보류한다.

```
10 ON KEY GOSUB 40, 60  ; [F1] 을 받으면 행번호 40, [F2] 를 받으면 행번호 60을 부르도록
                          정의한다.
20 KEY(1) ON:KEY(2) ON  ; [F1], [F2] 를 받으면 각각 지정된 서브루틴을 부르도록 허가한다.
30 GOTO 30              ; [CTRL] + [STOP] 을 받을 때까지 프로그램을 계속 실행시킨다.
40 PRINT "F1을 눌렀읍니다"; [F1] 을 눌렀다고 화면에 표시한다.
50 KEY(1) OFF:RETURN    ; [F1] 을 받아도 서브루틴을 부르지 않도록 금지시킨다.
60 PRINT "F2를 눌렀읍니다"; [F2] 를 눌렀다고 화면에 표시한다.
70 RETURN
```

● 참 조 : ON KEY GOSUB

KEY ON/OFF (키온/오프)

기능키의 내용을 화면에 표시할 것인지의 여부를 결정한다.

●문형식 : KEY ON
KEY OFF

●설 명 : ○ KEY ON을 실행하면, 기능키의 내용을 화면의 맨아래에 표시한다.
보통 상태에서는 [F1] ~ [F5] 를, [SHIFT] 와 함께 누르면 [F6] ~ [F10] 을 표시한다.
○ KEY OFF를 실행하면 기능키의 내용을 표시하지 않는다.
○ 시스템이 리세트되면 KEY ON을 실행한 상태로 된다.

●참 조 : KEY, KEY LIST

KILL (킬)

디스켓에 있는 화일을 지운다.

●문형식 : KILL"[〈장치명 : 〉] 〈화일명〉"

●설 명 : ○ 〈장치명 : 〉과 〈화일명〉에 대해서는 해설편의「화일」을 참조하십시오.
○ OPEN 되어 있는 화일을 지우려고 하면, "File already open" (화일 얼레디 오픈) 에러가 발생한다.

KILL "A : TEST. BAS" ; 드라이브A에 있는「TEST. BAS」라는 화일을 지운다.

LEFT$ (레프트스트링)

문자식의 왼쪽부터 문자열을 얻는다.

●문형식 : LEFT$ (〈문자식〉, 〈수식〉)

●설 명 : ○ 〈문자식〉의 왼쪽부터 〈수식〉으로 지정한 길이의 문자열을 얻는다.
○ 〈수식〉의 범위는 0～255이다.
○ 〈수식〉이 〈문자식〉의 총 문자수 보다 클 때는, 문자식 전체가 문자열로 되며, 〈수식〉이 0이면, 널 스트링이 된다.

A$="DAEWOO COMPUTER" 일 때,
B$=LEFT$(A$, 6) ; B$에는 A$의 왼쪽부터 6문자인「DAEWOO」가 대입된다.

●참 조 : RIGHT$

LEN (렝스)

문자열의 문자 갯수를 얻는다.

●문형식 : LEN (〈문자식〉)

●설 명 : ○ 〈문자식〉의 문자 갯수를 얻는다.
○ LEN은 제어 코드(CONTROL CODE)와 공백도 문자로 센다.

A$="DAEWOO MSX" 일 때,
A=LEN(A$) ; A에는 문자열 A$의 문자 갯수인 10이 대입된다.

LET (렛)

변수에 값을 대입한다.

● **문형식** : 〔LET〕〈변수명〉=〈식〉

● **설　명** : ○ 보통 LET을 생략한 상태로 사용한다.
○ 〈변수명〉은 첫 글자가 영문자인 영숫자이며, 예약어를 포함해서는 안된다.
또 2문자까지 가능하며, 초과되었을 경우 나머지는 무시된다.
○ 문자식을 수치변수에 대입하거나, 수치식을 문자변수에 대입하여 좌변과 우변의 형이 일치되지 않을 경우에는 "Type mismatch" (타잎 미스매치) 에러가 발생된다.

```
LET A=10     ;A에는 10이 대입된다.
B$="MSX"     ;B$에는「MSX」가 대입된다.
```

● **참　조** : 자료편의「예약어표」

LINE (라인)

그래픽 화면에 직선이나 사각형을 그린다.

● **문형식** : LINE〔〔STEP〕(X1, Y1)〕-〔STEP〕(X2, Y2)〔,〈색번호〉〕|〔,B〕 / 〔,BF〕|〔,〈논리연산기호〉〕

● **설　명** : ○ 시작점 (X1, Y1)과 끝점 (X2, Y2)을 연결하여 직선이나 사각형을 그린다.
○ X축은 SCREEN 2~5와 8에서는 0~255, SCREEN 6, 7에서는 0~511을 나타낼 수 있다.
○ Y축은 SCREEN 2~4에서는 0~191, SCREEN 5~8에서는 0~211을 나타낼 수 있다.
○ 〔STEP〕을 지정하면 최종 참조점부터 지정한 값 만큼 이동하여 점을 표시한다.
최종 참조점이란, 이전의 그래픽 명령에서 제일 마지막으로 지정된 점이다.
○ (X1, Y1)을 생략하고, -(X2, Y2)만을 지정하면, 최종 참조점부터 (X2, Y2)까지를 연결한 선을 그린다.
○ 〈색번호〉를 지정하면, 지정한 색으로 선이나 사각형을 그린다.
생략하면, 이전에 COLOR 명령으로 지정된 표시색이 사용된다.
(색번호에 대해서는 COLOR 명령을 참조하십시오)
○ 〔,B〕와 〔,BF〕를 생략하면 직선이 그려진다.
〔,B〕를 지정하면 시작점과 끝점을 대각으로 하는 사각형을 그리며 〔,BF〕를 지정하면, 사각형 안을 지정한 색으로 칠한다.
○ 〈논리연산기호〉는 SCREEN 5~8에서만 가능하며, 이것은 지정한 색과 바탕색을 다음과 같이 조작하는 것이다. (논리연산기호는 해설편의「색상」을 참조하십시오.)
지정한 색을 C, 화면의 바탕색을 SC로 할 때,

AND →C AND SC　　　PSET→C
OR　→C OR SC　　　PRESET→NOT(C)
XOR →NOT(C) AND SC OR C AND NOT(SC)

```
10 SCREEN 5
20 LINE (100, 100)—(150, 150), 8, BF;시작점 (100, 100)과 끝점 (150, 150)을 대각으로 하는
                                      빨간색의 사각형을 그리고, 그 안을 빨간색으로 칠한다.
30 LINE (80, 50)—(180, 150), 15      ;시작점과 끝점을 이어주는 하얀 선을 그린다.
40 LINE—(200, 180), 10               ;행번호 30의 끝점 (180, 150)을 시작점으로 하여
                                      점 (200, 180)을 잇는 노란선을 그린다.
50 LINE—STEP (30,—30), 13            ;시작점 (200, 180)과 끝점 (200+30, 180—30)을 이어주는
                                      보라색 선을 그린다.
60 GOTO 60
```

● 참　조 : SCREEN, COLOR

MAXFILES (맥스 화일즈)

사용할 화일의 수를 지정한다.

● 문형식 : MAXFILES=〈화일의 수〉

● 설　명 : ○ OPEN문으로 사용할 화일의 수를 정의한다.
○ 〈화일의 수〉는 0부터 15까지의 값을 지정할 수 있다.
○ 이 명령을 실행하면 변수가 초기화되고, 열어 놓은 화일은 닫힌다.
또 이전의 변수형 선언은 무효로 된다.
○ 「MAXFILES=0」라고 지정하면, SAVE 및 LOAD 명령만 실행할 수 있다.

```
MAXFILES=2  ;OPEN문으로 지정할 화일의 수를 2개로 지정한다.
```

● 참　조 : CLEAR, OPEN

MERGE (머어지)

화일에 있는 아스키 형식의 프로그램을 혼합한다.

● 문형식 : MERGE"[〈장치명 :〉] [〈화일명〉]"

● 설　명 : ○ 메모리상의 프로그램에 지정한 화일을 혼합하여 하나의 프로그램을 만든다.
○ 〈장치명 :〉에 대해서는 해설편의「화일」을 참조하십시오.
○ 지정된 〈화일명〉의 프로그램은 아스키형식으로 저장되어 있어야 한다.
○ 〈화일명〉은 〈장치명 :〉이 "CAS :"일 경우 생략할 수 있다. 생략할 경우는, 처음으로 찾은 아스키형식의 프로그램을 읽어들인다.
○ 화일중의 프로그램과 메모리안의 프로그램에 같은 행번호가 있는 경우에는 화일중의 행번호가 사용되므로, 행번호가 같지 않도록 주의해야 한다.

```
MERGE "CAS:TEST"  ;카세트테이프에 있는「TEST」라는 프로그램을 메모리의 프로그램과 합쳐
                   하나의 프로그램으로 만든다.
```

● 참　조 : SAVE, LOAD

LINE INPUT (라인인풋)

키보드로부터 입력된 1행 전체의 문자열을 문자변수에 대입한다.

● 문형식 : LINE INPUT〔"〈프롬프트문〉";〕〈문자 변수명〉

● 설　명 : ○ 이 명령을 실행하면, 프로그램은 키보드로부터 입력을 기다리며, 이때부터 [RETURN] 을 누를때까지, 입력된 문자열을 문자 변수에 대입한다.
○ 키보드로부터 입력되는 문자열의 길이는, 255문자 이내이다. 쉼표(,)나 인용부호(")도 데이타로 입력할 수 있다.
○ 〈프롬프트문〉을 지정하는 방법은 INPUT문과 마찬가지이다.

LINE INPUT "DATA"; A$　; 화면에 「DATA」라는 메시지를 표시하고 키입력을 기다린다.

| 참고 | [RETURN] = CHR$(13)+CHR$(10)=캐리지리턴+라인피드 |

● 참　조 : INPUT, LINE INPUT #

LINE INPUT # (라인인풋 넘버)

지정한 화일로부터 1행 전체의 데이타를 읽어들여 문자변수에 대입한다.

● 문형식 : LINE INPUT #〈화일번호〉,〈문자 변수명〉

● 설　명 : ○ 미리 OPEN문에서, INPUT모드로 지정된 화일로부터 캐리지리턴+라인피드로 구분되는 한행 단위(255문자 이내)의 데이타를 읽어들여 지정한 문자변수에 대입한다.
○ SAVE문으로 보존되는 ASCII(아스키) 형식의 프로그램 화일에서는 각 행이 캐리지리턴코드(CHR$(13))+라인피드코드(CHR$(10))로 구분되어 저장되므로, LINE INPUT # 명령으로 읽어들일 수 있다.

LINE INPUT # 1, A$; 화일번호1로 열려있는 화일에서 한행 단위의 데이타를 읽어들여 A$에 대입한다.

● 참　조 : OPEN, CLOSE, PRINT #, LINE INPUT

LIST/LLIST (리스트/라인리스트)

프로그램을 화면 또는 프린터에 출력한다.

● 문형식 : 〔L〕LIST〔〈행번호〉〕〔−〔〈행번호〉〕〕

● 설　명 : ○ 메모리에 있는 프로그램을 LIST는 화면에 표시하고, LLIST는 프린터에 출력한다.
○ 〈행번호〉대신에 . (마침표)를 사용하면, 베이직이 현재 가리키고 있는 행을 나타낸다. 즉, 에러가 발생되었을 때는 에러가 발생된 행을 지정한다.
○ LIST 기능을 잠시 중단하려면, [STOP] 을 누르고, 다시 [STOP] 을 누르면 LIST를 계속한다.
○ LIST 기능을 도중에서 끝내려면, [CTRL] + [STOP] 을 누른다.

LIST　; 프로그램의 처음부터 끝까지를 화면에 표시한다.
LIST 50　; 행번호 50의 행만 화면에 표시한다.
LIST −100　; 프로그램의 처음부터 행번호 100까지를 표시한다.
LIST 200−　; 프로그램의 행번호 200부터 끝까지 표시한다.
LIST 100−200; 프로그램의 행번호 100부터 행번호 200까지 표시한다.
LLIST　; 프로그램의 처음부터 끝까지를 프린터에 출력한다.

LOAD (로드)

프로그램을 읽어들인다.

● **문형식** : LOAD"[〈장치명 :〉] [〈화일명〉]"[,R]

● **설　명** : ○ 〈장치명 :〉은 데이타 레코더일 경우 "CAS:", 디스크 드라이브일 경우 "A:" 또는 "B:", 메모리 디스크일 경우 "MEM:" 또는 "MEM 0:"로 지정한다. 데이타 레코더만 연결되어 있는 경우에는 "CAS:"을 생략할 수 있다. 현재 사용중인 디스크 드라이브를 지정할 때는 〈장치명 :〉을 생략할 수 있다.
○ 〈화일명〉은 SAVE 명령으로 보존되어 있는 화일의 이름이다. 자세한 것은 해설편의 「화일」을 참조하십시오.
○ ,R을 지정하면 프로그램을 읽어들인 후, 곧바로 실행시킨다.
○ 이 명령은 데이타 레코더에 있는 화일을 읽어들일 경우 아스키형식의 프로그램을 읽어들인다. CSAVE로 보존되어 있는 내부어형식의 프로그램은 읽어들일 수 없다.

LOAD "CAS:TEST", R ; 카세트 테이프에 SAVE명령으로 보존된 「TEST」라는 이름의 프로그램을 읽어들인 후 실행시킨다.
LOAD "B:PROG" ; 드라이브B에 있는 디스켓 화일중 「PROG」라는 프로그램을 메모리로 읽어들인다.
LOAD "MEM:TEST" ; 메모리 디스크에 있는 「TEST」라는 화일을 읽어들인다.

● **참　조** : SAVE, CSAVE, CLOAD

LOC (엘오씨)

화일 중의 현재의 레코드 위치를 나타낸다.

● **문형식** : LOC (〈화일번호〉)

● **설　명** : ○ 〈화일번호〉는 OPEN문으로 지정한 것과 같아야 한다.
○ 랜덤화일에 대하여 이 함수를 사용하면 GET 또는 PUT문에 의해 읽혀지거나 쓰여진 후에 레코드의 번호를 얻는다. 화일을 OPEN한 후, 입출력을 행하지 않았을 경우 이 함수는 0을 얻는다.
○ 시퀀셜화일에 대하여 이 함수를 사용하면 그 화일을 OPEN한 후, 읽거나 쓰여지기 전의 레코드 수를 얻는다. 즉, 화일이 시퀀셜 입력 모드로 OPEN 되었을 경우에는 시스템이 최초의 섹터를 읽으므로 그 화일을 읽기전에도 이 함수의 값은 1로 된다.

A=LOC(1) ; 화일번호1로 열려있는 화일에서 읽혀지거나 써지는 레코드수가 변수 A에 대입된다.

LOCATE (로케이트)

커서의 위치를 지정한다.

● 문형식 : LOCATE〔〈X좌표〉〕〔,〈Y좌표〉〕〔,〈커서스위치〉〕
0 : 커서는 키(key) 입력을 기다릴 때 이외는 표시되지 않습니다.
1 : 커서는 항상 표시됩니다.

● 설 명 : ○ 텍스트 화면에서만 사용 가능하다.
○ 〈X좌표〉나 〈Y좌표〉가 화면의 범위(X축은 SCREEN 0, 9일때 40 또는 80, SCREEN 1일때 32이며, Y축은 24이다)를 넘게되면 각각의 최대치를 지정한 것으로 간주한다.
○ 〈커서스위치〉는 화면상의 커서의 표시여부를 결정하는 스위치입니다.
○ 이 명령은 PRINT문에서 표시하는 문자나, INPUT문에서 표시하는 프롬프트의 시작위치를 지정할 때 등에 쓰인다.

```
10 LOCATE 10, 12;커서를 X좌표 10, Y좌표 12의 위치에 놓는다.
20 PRINT "MSX" ;지정한 위치에 「MSX」를 표시한다.
```

LOF (엘오에프)

지정된 화일의 크기를 나타낸다.

● 문형식 : LOF(〈화일번호〉)

● 설 명 : ○ 〈화일번호〉로 지정한 화일은 OPEN문에서 입력용 또는 출력용으로 열어놓아야만 한다.
○ 화일의 크기를 바이트 단위로 구한다.

```
A=LOF(1)   ;A에는 화일번호 1로 열려있는 화일의 크기가 대입된다.
```

LOG (로그)

자연대수를 얻는다.

● 문형식 : LOG(〈수식〉)

● 설 명 : ○ 〈수식〉에서 지정하는 값의 자연대수(e를 밑수로 하는 대수)를 구한다.
○ 〈수식〉은 어떤형이라도 상관없으나 0보다 커야한다.
○ 결과는 배정도 실수형이다.

```
A=LOG(10)   ;A에는 10의 자연대수인 2.302585092994가 대입된다.
```

LPOS (라인포지션)

프린터의 헤드 위치를 조사한다.

● 문형식 : LPOS(〈인수〉)

● 설 명 : ○ LPOS에 의하여 얻은 값은, 프린터 버퍼상의 헤드의 위치로서 반드시 실제의 헤드 위치와 일치하는 것은 아니다.
○ 〈인수〉는 더미(DUMMY)로서 어떤것이 있어도 관계없다.

```
A=LPOS(0)   ;A에는 현재 프린터의 헤드 위치가 대입된다.
```

● 참 조 : LPRINT

LPRINT (라인프린트)

프린터에 문자열이나 수치를 출력한다.

●**문형식** : LPRINT〔〈식〉……〕

●**설　명** : 이 명령은 프린터로 출력된다는 것을 제외하면, 그 사용방법이 PRINT 명령과 같으므로, 그것을 참조하십시오.

```
10 A$= "ABC"          ;A$에 문자열「ABC」를 대입한다.
20 LPRINT A$;"DEF"   ;A$의 내용과 문자열「DEF」를 프린터에 출력한다.
```

●**참　조** : PRINT

LPRINT USING (라인프린트유싱)

문자열이나 수치를 지정한 서식으로 프린터에 출력한다.

●**문형식** : LPRINT USING "〈서식〉" ; 〈식〉…

●**설　명** : ○ 인용부호(")로 둘러싸인 〈서식〉에 따라, 지정된 문자열이나 수치를 편집하여 그 결과를 프린터에 출력한다.
○ 이 명령은 프린터로 출력된다는 것을 제외하면, 그 사용방법이 PRINT USING 명령과 같으므로 그것을 참조하십시오.

```
10 A$="ABC"                    ;A$에 문자열「ABC」를 대입한다.
20 LPRINT USING "@DEF";A$;A$의 내용을 @의 위치에 놓고 프린터에 출력한다.
```

●**참　조** : PRINT USING

LSET (레프트셀)

랜덤 화일 버퍼에 데이타를 왼쪽부터 써넣는다.

●**문형식** : LSET〈문자변수명〉=〈문자식〉

●**설　명** : ○ 이 명령은 PUT명령(랜덤화일 버퍼에 있는 데이타를 지정한 레코드에 써넣는다)을 하기위한 준비단계이다.
○ 〈문자식〉의 길이가 FIELD문으로 지정된 문자변수의 길이보다 짧을 경우, 남은 공간은 공백으로 채워진다.
○ 〈문자식〉의 길이가 FIELD문으로 지정한 길이보다 길면 〈문자식〉의 뒷 부분이 잘린다.
○ FIELD문으로 정의되지 않은 문자변수에 대해서도 이 명령을 사용할 수 있다. 이것은 PRINT 출력의 구성에 이용할 수 있다.

```
100 A$=SPACE$(28)    ;A$을 28자의 길이로 지정
110 LSET A$ = N$     ;A$에 N$ 의 내용을 왼쪽부터 써넣는다.
```

○ 수치 데이타는 이 명령을 실행하기 전에, 미리 문자형으로 바꾸어야 한다. (MKI$, MKS$, MKD$ 명령을 참조하십시오.)

```
LSET A$=MKS$(AMT)   ;수치변수의 값을 문자형으로 바꾸어 버퍼의 왼쪽부터 써넣는다.
```

MID$ (미들스트링) ①

문자열 중에서 지정한 길이의 문자열을 얻는다.

● **문형식** : MID$ (<문자열>, <식1> [, <식2>])

● **설　명** : ○ <문자열>의 왼쪽에서 <식1>번째의 문자부터 <식2>만큼의 문자열을 얻는다.
○ <식1>의 범위는 1～255이고, <식2>의 범위는 0～255이다.
○ <문자열>의 문자수가 <식1>보다 작을 경우는 널스트링을 얻는다.
○ <식2>를 생략한 경우와 <문자열>의 <식1>번째 부터 오른쪽에 있는 문자수가 <식2>보다 작을 경우에는 <문자열>의 <식1>번째부터 오른쪽에 있는 모든 문자열을 얻는다.

```
10 A$="DAEWOO MSX COMPUTER";A$에 문자열을 대입한다.
20 PRINT MID$(A$, 8, 3)     ;A$에 대입된 문자열의 왼쪽에서 8번째 자리부터 3문자
                             즉,「MSX」를 화면에 표시한다.

A$=MID$("COMPUTER", 6, 5)   ;문자열의 6번째 자리부터의 문자길이가 5보다 작으므로
                             6번째 문자부터 오른쪽의 모든 문자(TER)를
                             A$에 대입한다.
B$=MID$("DAEWOO", 4)        ;문자열의 4번째 문자부터 오른쪽의 모든 문자를
                             변수 B$에 대입한다.
```

● **참　조** : LEFT$, RIGHT$, MID$

MID$ (미들스트링) ②

문자열의 일부를 바꾸어 넣는다.

● **문형식** : MID$ (<문자변수>, <식1> [, <식2>]) =<문자열>

● **설　명** : ○ <문자변수>의 <식1>번째 문자부터 <식2>개 문자를 <문자열>의 처음부터 <식2>개의 문자열로 바꿔넣는다.
○ <식1>의 값은 <문자변수>의 길이보다 크거나 0이하가 되지 않도록 해야한다.
○ <식2>를 생략하거나 <문자열>의 문자수보다 많이 지정한 경우는 <문자열>의 모든 문자와 대치된다.
○ <문자변수>의 <식1>번째부터 오른쪽에 있는 문자수가 <식2>보다 작을 경우에는 <문자변수>의 <식1>번째부터 오른쪽에 있는 모든 문자열의 길이를 <식2>로 간주하고 그길이 만큼을 <문자열>과 바꾸어 넣는다.

```
A$="ABCD":MID$(A$, 3, 2)="XY":PRINT A$
  ;「ABCD」의 3번째부터 2문자를「XY」로 바꾼다. 즉, A$에는「ABXY」가 대입된다.
C$="MSX":MID$(C$, 2, 3)="ISS":PRINT C$
  ;C$에 대입된「MSX」의 2번째부터의 문자길이가 2개밖에 되지 않으므로, 문자열「ISS」중에서
   처음 2문자「IS」만 바꿔놓는다. 즉, C$에 는「MIS」가 대입된다.
```

● **참　조** : MID$①

MKI$/MKS$/MKD$ (메이크 인테저 스트링/싱글/더블)

수치 데이타를 문자형으로 변환시킨다.

● 문형식 : MKI$(〈정수형 수치 데이타〉)
MKS$(〈단정도형 수치 데이타〉)
MKD$(〈배정도형 수치 데이타〉)

● 설 명 : ○ 랜덤 화일에 기록될 데이타는 문자형이어야 하므로, 수치 데이타를 먼저 문자형으로 만들어야 한다.
○ MKI$은 정수형 수치를 2바이트 문자열로, MKS$은 단정도형 수치를 4바이트 문자열로, MKD$은 배정도형 수치를 8바이트 문자열로 각각 변환한다.

```
10 OPEN "A:NUMBER. DAT" AS #1                ;「NUMBER. DAT」라는 화일을 연다.
20 FIELD #1, 2 AS IN$, 4 AS SG$, 8 AS DB$;IN$를 2자리, SG$를 4자리, DB$를 8자리로
                                              정의한다.
30 READ A%, B!, C#                           ;A%에 5, B!에 3.123, C#에 12.3456789를
                                              대입한다.
40 LSET IN$=MKI$(A%)                         ;IN$에 A%의 값을 문자형으로 바꾸어 대입한다.
50 LSET SG$=MKS$(B!)                         ;SG$에 B!의 값을 문자형으로 바꾸어
                                              대입한다.
60 LSET DB$=MKD$(C#)                         ;DB$에 C#의 값을 문자형으로 바꾸어
                                              대입한다.
70 PUT #1, 1                                 ;열려있는 화일의 레코드번호 1에 데이타를
                                              써넣는다.
80 CLOSE #1                                  ;화일을 닫는다.
90 END
100 DATA 5, 3.123, 12.3456789
```

● 참 조 : CVI, CVS, CVD, LSET, RSET

MOTOR (모우터)

데이타 레코더의 동작을 제어한다.

● 문형식 : MOTOR ON
MOTOR OFF
MOTOR

● 설 명 : ○ ON을 지정하면 모우터가 동작하고, OFF를 지정하면 모우터를 정지시킨다.
○ MOTOR로만 지정하면, 현재 모우터가 ON이면 OFF로, OFF이면 ON으로 한다.
○ 이 명령은 리모트단자가 있는 데이타 레코더를 연결하였을때, 모우터의 동작을 제어하기 위한 명령이다.

NAME (네임)

디스켓상의 화일 이름을 변경한다.

● 문형식 : NAME "〈구 화일명〉" AS "〈신 화일명〉"

● 설　명 : ○ 〈구 화일명〉은 디스켓 상에 존재해야 하고, 〈신 화일명〉은 존재하고 있지 않아야 된다.
　　○ 이 명령을 실행하면, 〈구 화일명〉이 있던 위치에 〈신 화일명〉이 놓여질 뿐, 화일의 크기나 존재하는 위치 등은 변화하지 않는다.

```
NAME "ACCTS" AS "LEDGER" ; 디스켓 상의 화일중 「ACCTS」라는 이름의 화일을
                           「LEDGER」이라고 변경시킨다.
```

NEW (뉴우)

메모리에 있는 프로그램을 지우고, 모든 변수를 초기화 한다.

● 설　명 : ○ 이 명령은 새로운 프로그램을 입력하기 전에 실행한다.
　　○ 이 명령을 실행하면, 메모리에 있는 프로그램은 모두 지워지고, 정의되어 있는 변수도 모두 초기화 되며, 열려있던 화일도 모두 닫힌다.

```
10 PRINT "PROGRAM" ; 「PROGRAM」을 화면에 나타낸다.
NEW                ; 프로그램을 지운다.
```

● 참　조 : DELETE, ERASE

OCT$ (옥탈스트링)

수식을 8진수 표기의 문자열로 변환한다.

● 문형식 : OCT$ (〈수식〉)

● 설　명 : ○ 〈수식〉의 범위는 −32768～65535이다.
　　○ 결과는 0～7의 숫자로 구성되며, 6자리까지 나타낼 수 있으나 맨앞부터 계속되는 0은 표시되지 않는다.

```
PRINT OCT$(100) ; 십진수 100에 대한 8진수의 값은 144이다.
```

● 참　조 : HEX$

ON ERROR GOTO (온 에러 고투)

에러가 일어날 때 실행할 행을 정의한다.

● 문형식 : ON ERROR GOTO 〈행번호〉

● 설 명 : ○ 먼저, 에러 처리루틴을 프로그램 안에 입력하여 놓고, 프로그램 맨앞에 이 명령으로 처리루틴의 처음 행번호를 지정하여 놓으면 에러가 일어날 때에, 그 루틴을 실행한다.

○ 에러 처리루틴의 마지막에 RESUME 명령으로 에러 회복처리를 하여야만 한다.

○ ON ERROR GOTO 명령은 다른 ON ERROR GOTO문, RUN, CLEAR, NEW를 실행할 때까지 유효하기 때문에, 그대로 프로그램을 끝내면 직접 모드에서 명령을 잘못 쳐넣었을때 에러 처리루틴으로 점프하여 프로그램이 실행된다.
따라서 프로그램이 끝나기 전에는 ON ERROR GOTO 명령을 무효로 하는, ON ERROR GOTO 0을 실행하여야 한다.

```
10 ON ERROR GOTO 60  ;에러가 발생하면 행번호 60으로 점프하라고 지정.
20 A$=INKEY$         ;키보드로 부터 키를 받아들여 변수 A$에 대입한다.
30 IF A$= "Y" THEN 80 ;받아들인 데이타가 Y이면 행번호 80으로 점프한다.
40 A%=A%+100         ;정수형 변수 A%에 100을 더한다.
50 GOTO 20           ;행번호 20으로 점프한다.
60 PRINT "Sorry":A%=0;정수형 변수의 값이 정수범위를 초과하면 이 행으로 점프하므로
                      「Sorry」를 화면에 표시하고 변수 A%에 0을 대입한다.
70 RESUME            ;에러 회복 처리를 한다.
80 ON ERROR GOTO 0   ;ON ERROR GOTO 명령을 무효로 한다.
```

● 참 조 : RESUME, ERL/ERR, ERROR

ON GOTO/ON GOSUB (온 고투/온 고서브)

조건에 따라 지정된 행으로 점프한다.

● 문형식 : ON 〈식〉 GOSUB 〈행번호〉 〔,〈행번호〉…〕
ON 〈식〉 GOTO 〈행번호〉 〔,〈행번호〉…〕

● 설 명 : ○ 〈식〉의 값에 대응하여, 지정된 행으로 점프한다.

○ ON GOTO는 지정한 행으로 점프하고, ON GOSUB는 지정한 행에서 시작되는 서브루틴을 호출한다.

○ 〈행번호〉는 왼쪽부터 〈식〉의 값인 1, 2, 3…순으로 대응한다.

```
ON A GOSUB 100, 200 ;변수A의 값이 1이면 행번호 100, 2이면 행번호 200으로 시작하는
                     서브루틴을 호출한다.
ON B GOTO 50, 70, 80 ;변수B의 값이 1이면 행번호 50, 2이면 행번호 70, 3이면 행번호 80으로
                      점프한다.
```

● 참 조 : GOTO, GOSUB

ON INTERVAL GOSUB (온인터벌 고서브)

시간에 의한 인터럽트 처리루틴의 시작행을 정의한다.

● **문형식** : ON INTERVAL=〈시간〉 GOSUB 〈행번호〉

● **설　명** : ○ 이 명령은 인터럽트가 걸리는 시간 간격을 설정하고, 인터럽트가 걸릴때에 호출할 서브루틴의 시작행을 정의한다.
○ 〈시간〉은 1/60초 단위이고, 1~65535의 값을 가질 수 있다.
○ 〈행번호〉는 인터럽트 처리루틴이 시작되는 행번호이다.
○ 이 명령은 인터럽트 정의를 할 뿐이고, INTERVAL ON 문을 실행하지 않는 한 인터럽트는 걸리지 않는다.

```
10 ON INTERVAL=600 GOSUB 40;600/60=10, 즉 10초마다 행번호 40에서 시작되는
                           서브루틴을 호출하도록 정의한다.
20 INTERVAL ON             ;행번호 10에서 정의한 것을 실행할 수 있게 한다.
30 GOTO 30                 ;자신의 행번호로 점프하도록 한다.
40 PRINT "*";              ;화면에 *를 표시한다.
50 RETURN                  ;인터럽트가 일어난 곳으로 돌아간다.
```

● **참　조** : INTERVAL ON/OFF/STOP

ON KEY GOSUB (온키고서브)

기능키에 의한 인터럽트 처리루틴의 시작행을 정의한다.

● **문형식** : ON KEY GOSUB〔〈행번호〉〕〔,〈행번호〉…〕

● **설　명** : ○ 이 명령은 기능키를 눌렀을때, 호출할 서브루틴의 시작행을 정의한다.
○ 〈행번호〉는 처리루틴이 시작되는 행번호이며 나열순서는 기능키의 번호와 1대1로 대응한다. 기능키가 [F1] 부터 [F10]까지 있으므로, 최대 10개의 〈행번호〉를 나열해서 쓸 수 있다.
○ 이 명령은 인터럽트 처리루틴을 지정할 따름이며, KEY(n) ON문을 실행함으로써, 기능키에 대한 인터럽트가 유효하게 된다.

ON KEY GOSUB 100, 200: 기능키 [F1]을 누르면 행번호 100, 기능키 [F2]를 누르면 행번호 200에서 시작하는 서브루틴을 호출하도록 정의한 것.

● **참　조** : KEY(n) ON/OFF/STOP

ON SPRITE GOSUB (온 스프라이트 고서브)

스프라이트가 서로 충돌할 때 호출될 서브루틴의 시작행을 정의한다.

●문형식 : ON SPRITE GOSUB〈행번호〉

●설 명 : ○ 이 명령을 정의하여 놓고, SPRITE ON을 실행하면, 화면에 나타난 스프라이트가 충돌할 때에 지정한 행을 호출한다.
○ 스프라이트의 충돌은 스프라이트 패턴 1과 1이 만날때 생긴다.
○ 행번호는 스프라이트의 충돌 인터럽트가 발생될 때에 처리되는 서브루틴의 첫행이다.
○ SCREEN 4～8의 스프라이트에서는 특수한 색 번호를 가지고 있기 때문에 충돌을 검출하지 않을 경우가 있다. 자세한 것은 COLOR SPRITE문을 참조하십시오.

ON SPRITE GOSUB 90 ;스프라이트 패턴이 겹칠 때에는 행번호 90에서 시작되는 서브루틴을 호출하라고 정의한 것.

●참 조 : SPRITE ON/OFF/STOP

ON STOP GOSUB (온 스톱 고서브)

CTRL + STOP 키가 눌러질 때, 호출될 서브루틴의 시작행을 정의한다.

●문형식 : ON STOP GOSUB〈행번호〉

●설 명 : ○ 이 명령을 선언한 후, STOP ON문이 실행되면 CTRL + STOP을 누를때에 지정한 행부터 시작되는 처리루틴을 실행한다.
○ 이 명령은 프로그램 실행중에 CTRL + STOP을 잘못 눌러서 프로그램의 실행이 중단되는 것을 방지하기 위해 사용한다. 그러므로 이 명령을 실행했을 경우는 CTRL + STOP의 기능이 상실되어 시스템 리세트를 행하지 않는한 프로그램을 중단시킬 수 없게된다.

ON STOP GOSUB 100 ; CTRL + STOP 키가 눌러지면, 프로그램의 실행은 행번호 100으로 이동하라고 정의한 것.

●참 조 : ON KEY GOSUB, STOP ON/OFF/STOP

ON STRIG GOSUB (온 스트릭 고서브)

조이스틱의 트리거 버튼이 눌러질 때에 호출될 서브루틴의 시작행을 정의한다.

● 문형식 : ON STRIG GOSUB 〔<행번호>〕〔, <행번호>…〕

● 설　명 : ○ 이 명령을 선언한 후, STRIG ON문이 실행되면 스페이스키 또는 조이스틱의 트리거 버튼을 누를때에, 지정한 행으로 시작되는 서브루틴이 실행된다.

○ <행번호>는 최대 5개까지 나열하여 지정할 수 있고, 순서대로 다음과 같이 대응한다.

1번째 : 키보드의 스페이스키
2번째 : 포트 1에 연결되어 있는 조이스틱의 트리거 버튼 1
3번째 : 포트 2에 연결되어 있는 조이스틱의 트리거 버튼 1
4번째 : 포트 1에 연결되어 있는 조이스틱의 트리거 버튼 2
5번째 : 포트 2에 연결되어 있는 조이스틱의 트리거 버튼 2

단, 조이스틱에 따라서는 트리거 버튼 2가 없는 것도 있다.

ON STRIG GOSUB, 100, 200 ; 포트1에 연결되어 있는 조이스틱의 트리거 버튼 1을 누르면 100행으로, 포트2에 연결되어 있는 조이스틱의 트리거 버튼 1을 누르면 200행으로 프로그램의 실행이 이동하도록 정의한 것.

● 참　조 : STRIG(n) ON/OFF/STOP, ON KEY GOSUB

OPEN (오픈)

지정한 화일을 연다.

● **문형식** : OPEN"〈장치명 :〉〈화일명〉" [FOR〈모드〉] AS#〈화일번호〉[LEN=〈레코더 길이〉]

● **설　명** : ○ 이 명령은 지정한 화일에 화일번호와 화일버퍼를 할당하고, 데이타를 입출력할 수 있도록 하는 명령이다.

○ 아래와 같은 명령을 실행하기 위해서는 미리 OPEN문을 실행해야 한다.

PRINT #
INPUT #
PRINT # USING
INPUT$
LINE INPUT #
GET
PUT

〈장치명〉은 다음과 같이 6가지가 있다.

장치명	입출력장치	장치명	입출력장치
CAS:	데이타 레코더	CRT:	텍스트 화면
A: 또는 B:	디스크 드라이브	GRP:	그래픽 화면
MEM: 또는 MEM 0:	메모리 디스크	LPT:	프린터

〈화일명〉은 〈장치명:〉이 CRT:, GRP:, LPT: 일 경우에는 지정하지 않는다.
디스크 드라이브를 지정할 때는 〈장치명:〉을 생략할 수도 있다.

○ 〈모드〉에는 다음의 4종류가 있다.

FOR INPUT —— 시퀀셜 화일로부터 입력할 것을 지시한다.
FOR OUTPUT —— 시퀀셜 화일에 출력할 것을 지시한다.
FOR APPEND —— 시퀀셜 화일뒤에 데이타를 추가할 것을 지시한다.
생략하면 —— 랜덤화일에 입출력 할 것을 지시한다.

○ INPUT 모드는 기억장치에 한해서 사용된다.
APPEND 모드는 외부 기억장치 내의 시퀀셜 화일에 한해서 사용된다.
(단, 데이타 레코더는 제외)

○ 〈화일번호〉는 1부터 15까지 사용할 수 있으나, MAXFILES문으로 지정한 수를 넘어서는 안된다.
이미 열려진 화일이 이용하고 있는 번호를 또 다시 쓸 수는 없다.

○ 〈레코더길이〉는 랜덤 입출력 모드로 화일을 OPEN 할 때, 최대 레코더길이를 결정하는 것이다. 1~256 범위의 값을 지정하며, 생략하면 256으로 결정된다.

```
10 OPEN "CRT:" FOR OUTPUT AS #1 ;텍스트화면에 써내보내는 화일을 연다.
20 PRINT #1, "FILE OPEN"        ;열려있는 화일에 「FILE OPEN」을 써낸다.
30 CLOSE                        ;열려있는 모든 화일을 닫는다.
```

● **참　조** : CLOSE, INPUT #, PRINT #, MAXFILES, INPUT$, EOF

4 베이직 설명서

OUT (아웃)

지정한 출력포트에 1바이트의 데이타를 보낸다.

● 문형식 : OUT〈포트번호〉, 〈식〉

● 설 명 : ○ 〈포트번호〉에 대해서는 자료편의「I/O맵」을 참조하십시오.
○ 이 명령은, 시스템 하드웨어와 밀접한 관계를 가지고 있으므로, 다른 MSX 기종에서는 다르게 동작할 경우도 있다.

OUT &HA8, &H10 ; I/O포트 &HA8은 슬롯을 지정하는 포트로써 &H10를 써넣으면 슬롯0과 1을 사용가능하게 설정한다.

● 참 조 : INP

PAD (패드)

TABLET(태블리트), 마우스, 전자펜의 상태를 조사한다.

● 문형식 : PAD(〈수식〉)

● 설 명 : ○ 〈수식〉은 0～23의 수치를 지정할 수 있으며 각각 다음과 같은 장치들이 대응된다.

0～3 : 태블리트1
4～7 : 태블리트2
8～11 : MSX 전용 전자펜
12～15 : 마우스1
16～19 : 마우스2
20～23 : 대우 MSX 전용 전자펜

○ 이 명령을 실행하면 〈수식〉에 따라 다음과 같은 결과를 얻는다.

눌러져 있으면 −1 그렇지 않으면 0	눌러진 점의 X좌표	눌러진 점의 Y좌표	스위치가 눌러져 있으면 −1	
PAD(0)	PAD(1)	PAD(2)	PAD(3)	Port1에 접속된 태블리트
PAD(4)	PAD(5)	PAD(6)	PAD(7)	Port2에 접속된 태블리트
PAD(8)	PAD(9)	PAD(10)	PAD(11)	전용 전자펜 카트리지
PAD(12)	PAD(13)	PAD(14)	PAD(15)	Port1에 접속된 마우스
PAD(16)	PAD(17)	PAD(18)	PAD(19)	Port2에 접속된 마우스
PAD(20)	PAD(21)	PAD(22)	PAD(23)	전자펜의 상태

X=PAD(1) ; 범용포트1에 접속된 태블리트 등이 눌러지면 그점의 X좌표를 변수X에 대입한다.

● 참 조 : STRIG, PDL

PAINT (페인트)

지정된 경계색으로 둘러싸인 부분을 지정한 〈영역색〉으로 칠한다.

● 문형식 : PAINT〔STEP〕(X, Y)〔, 〈영역색〉〕〔, 〈경계색〉〕

● 설 명 : ○ (X, Y)에서 지정한 좌표가 포함된 〈경계색〉으로 정해진 영역을 지정한 〈영역색〉으로 칠한다.
○ STEP을 붙이면 최종 참조점의 상대 좌표로 된다.
○ 〈영역색〉을 생략한 경우에는, COLOR문에서 지정한 표시색이 사용된다.
○ 〈경계색〉을 생략한 경우에는, 〈영역색〉과 같은색이 설정된다.
○ SCREEN 2, 4에서는 〈영역색〉과 〈경계색〉이 같아야 한다.

주의 ○ 이 명령이 지정한 경계색(SCREEN 2, 4에서는 영역색)으로 둘러싸인 영역이 없으면 화면 전체가 칠해진다.
○ 이 명령은 그래픽 모드에서만 사용할 수 있다.

```
10 SCREEN 5              ;화면을 그래픽 모드로 한다.
20 CIRCLE(128, 96), 50, 11;칼라번호 11의 색으로 중점(128, 96), 반지름 50인 원을 그린다.
30 PAINT(128, 96), 6, 11 ;점(128, 96)을 포함하여 경계색(11)으로 둘러싸인 부분을
                          진한 적색(6)으로 칠한다.
40 GOTO 40               ;화면을 계속 그래픽 상태로 한다.
```

● 참 조 : SCREEN, COLOR

PDL (패들)

PADDLE(패들)의 상태를 조사한다.

● 문형식 : PDL(〈패들번호〉)

● 설 명 : ○ 〈패들번호〉는 1~12의 값을 지정할 수 있다. 값이 기수(1, 3, 5, 7, 9, 11)이면 포트1에 연결된 패들이 지정되고, 우수(2, 4, 6, 8, 10, 12)이면 포트2에 연결된 패들이 지정된다.
○ 패들의 상태는 회전의 정도에 따라 0~255 사이의 값을 얻는다.

```
X=PDL(2);범용포트2에 연결된 패들의 상태를 조사하여 그 결과를 X에 대입한다.
```

● 참 조 : PAD, STRIG

PEEK (피크)

메모리내의 지정된 번지의 내용을 읽어낸다.

● 문형식 : PEEK (〈번지〉)

● 설　명 : ○ 〈번지〉는 −32768～65535(&H0～&HFFFF)의 값을 갖는 수식으로 지정한다.
　○ 읽어내는 값은 0～255의 값(1바이트)이다.

```
A=PEEK(&HA000)    ;메모리번지 &HA000의 내용을 읽어 A에 대입한다.
```

● 참　조 : POKE, 자료편의 「메모리맵」

PLAY (플레이) ①

음악을 연주하고 있는가를 조사한다.

● 문형식 : PLAY (〈채널〉)

● 설　명 : ○ 각 채널에서 지정한 음악을 연주하고 있는가를 조사하여, 연주중이면 −1, 그렇지 않으면 0을 얻게된다.
　○ 〈채널〉은 0～3의 값으로 지정할 수 있으며, 각 채널과의 대응은 다음과 같다.
　　0: 채널1, 2, 3 (어느쪽이라도 연주중이면 −1의 값을 얻는다)
　　1: 채널1
　　2: 채널2
　　3: 채널3

```
10 A$="O4L4T120V4"      ;옥타브, 음의 길이, 속도, 음량을 지정한다.
20 PLAY A$              ;A$을 1개의 채널로 지정한다.
30 PLAY "C", "E", "G"   ;A채널의 음정을 지정한다.
40 IF PLAY(0)=0 THEN 30;채널연주가 모두 끝났는지를 조사한다.
50 GOTO 40              ;연주가 아직 끝나지 않았으므로 40행으로 다시간다.
```

PLAY (플레이) ②

음악을 연주한다.

● 문형식 : PLAY 〈문자식1〉 〔, 〈문자식2〉〕 〔, 〈문자식3〉〕

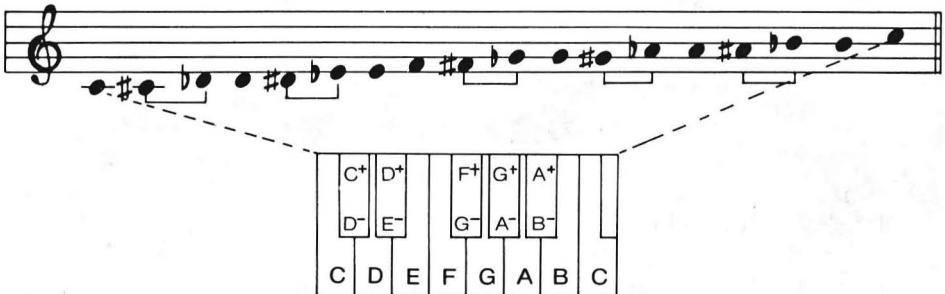
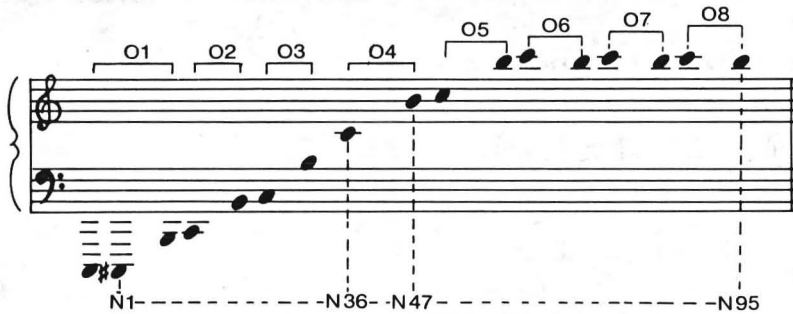
● 설 명 : ○ 〈문자식1〉, 〈문자식2〉, 〈문자식3〉은 각각 VOICE 채널 1, 2, 3에 대응하므로, 최대 3중화음까지 낼 수 있다.
○ 〈문자식〉이 널스트링이면 해당 VOICE 채널은 어떤 음도 내지 않는다.
○ 〈문자식〉은 아래에 서술된 음악 마크로 명령을 한 개이상 나열하며, 문자정수, 문자변수 또는 그것들을 조합한 식이다.
○ 음악 마크로 명령은 음의높이, 길이, 속도, 크기, 음색, LOCAL 마크로의 6종류가 있다.

```
PLAY "CDEFGABO5C"
PLAY A$+B$, C$        ;음정을 변수명으로 지정한 후 변수명을 사용하여 여러종류의 음을 낸다.
```

＊음의높이

O〈수치〉: 값을 1～8의 수치로 지정한다. 생략하면 O4로 지정된다. 한번 옥타브를 지정하면, 다음 옥타브를 지정할 때까지 유효하다.
예) PLAY "O3CEGO4" ;옥타브3의 도미솔을 낸후 옥타브4로 다시 변환한다.

A～G : 음의 높이를 음계명(도, 레, 미, 파, 솔, 라, 시)으로 표현한다.
C～G가 다장조의 「도」～「솔」, A와 B가 「라」와 「시」에 각각 대응한다.
A～G의 뒤에 ＃ 혹은 ＋를 붙이면 반음이 올라가고 반대로 －를 붙이면 반음이 내려간다.
예) PLAY "CE＃GEF－" ;미(E)음에서 반음 올리고, 파(F)음에서 반음 내린다.

N〈수치〉: 음의 높이를 0～96의 값으로 표시한다.
「N1」은 「O1」의 C＃과 같은 높이이고, N2, N3, …로 반음씩 올라가며, N96은 O8의 B와 같게 된다.
N0는 쉼표(R)와 같이 동작된다.
예) PLAY "N36N37N38N39N40N41N42N43" ;N36～N43에 해당하는 음정을 소리낸다.

＊음의길이

L〈수치〉: 음의 길이를 1/〈수치〉음으로 한다. 〈수치〉의 범위는 1～64이다.
L1은 온음(4박자), L2는 2분음(2박자), …, L64는 64분음(1/16박자)이다.
초기값은 L4(한박자)이다.
한번 음의 길이를 지정하면 다음 음의 길이가 지정될 때까지 유효하다.
한음에 대해서만 길이를 변화하고 싶을 때는 음계명 뒤에 음의 길이를 표시하는 값을 첨가한다. 예를들면 L16C는 C16과 같다.
예) PLAY "L4DCFG8O5" ;솔(G)음의 길이는 8이다.

R〈수치〉: 쉼표를 표시한다.
〈수치〉의 지정방법은 L명령과 같다.
〈수치〉를 생략하면, R4로 된다.
이 명령은 L명령에서 지정한 길이에 영향을 미치지 않는다.

•(점) : 음계명 또는 쉼표뒤에 붙여 본래 음의 길이를 1.5배로 한다.
2개를 나열해 쓰면 바로 앞음길이의 1.5배의 1.5배가 된다.

＊속도

T〈수치〉: 템포를 지정한다.
〈수치〉는 1분간에 연주하는 4분음표의 수를 지정한다.
〈수치〉의 범위는 32～255이다.
초기치는 120인데 한번 속도를 지정하면 다음 속도를 지정할때까지 유효하다.

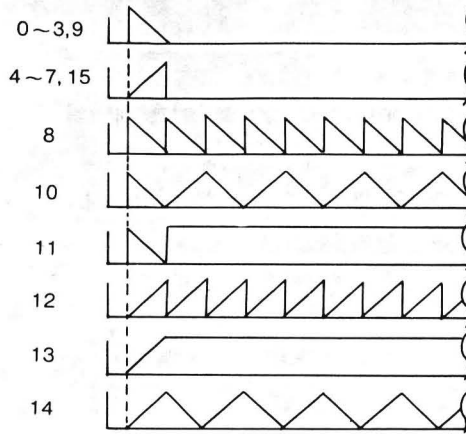
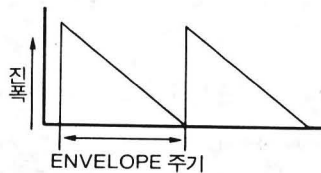
＊음의크기

V〈수치〉: 음량을 지정한다.
〈수식〉의 범위는 0～15이고, 값이 클수록 음도 크다.
한번 음량을 지정하면, 다음 음량을 지정할 때까지 유효하다.

＊음색

음색은 ENVELOPE 주기(M마크로)와 ENVELOPE 형상(S마크로)으로 결정된다.

S〈수치〉: ENVELOPE 형상(음량변화의 파형)을 지정한다.
수치와 파형은 다음과 같이 지정하는데, S마크로는 V마크로와 동시에 지정할 수 없다.

S의 값에 대한 ENVELOPE 형상

M〈수치〉: ENVELOPE 주기를 정한다. ENVELOPE 주기는 0～65535의 범위를 지정하고, 이 범위를 넘으면 ENVELOPE 형상의 특징이 나오지 않는다.

ENVELOPE 형상과 ENVELOPE 주기를 조합해서 여러가지 음을 낼 수 있다.

예) PLAY "V9CDE"
PLAY "S9M3000CDE"
PLAY "S4M2000CDE"
PLAY "S8M700CDE"
PLAY "S13M300CDE"
} 도레미(CDE)의 음을 여러 형태로 바꾸어 준다.

*LOCAL 마크로

X〈문자변수〉: 〈문자변수〉중의 문자열에 포함되어 있는 음악 명령을 실행한다.
〈문자변수〉에는 음악 마크로를 먼저 넣고, 뒤에는 세미콜론(;)을 반드시 써야 한다.

예) A$="CCGGAAG": B$="FFEEDDC"
PLAY "XA$;XB$;"

음을 조금씩 변화해 가면서 반복하는 연주를 할 때 편리하다.

수치변수지정: 다음 형식으로 수치변수도 지정할 수 있다.

=수치변수;

예) 10 FOR J=1 TO 64
20 PLAY "L=J;CDE"
30 NEXT
} 도레미(CDE)음이 점점 템포가 빨라 지면서 64번을 소리낸다.

POINT (포인트)

지정된 점의 색을 조사한다.

● 문형식 : POINT(X, Y)

● 설　명 : ○ 지정된 점의 색번호를 얻는다.
　　○ 스프라이트 패턴이 표시되어 있어도, 스프라이트 패턴의 색은 조사할 수 없다.
　　이 경우는 바탕색이 얻어진다.

A=POINT(128, 96)　;점(128, 96)의 색상의 값을 A에 대입한다.

● 참　조 : COLOR

POKE (포크)

메모리에 데이타를 써넣는다.

● 문형식 : POKE 〈번지〉, 〈데이타〉

● 설　명 : ○ 〈번지〉는 −32768∼65535 범위(&H0∼&HFFFF)이며, 〈데이타〉는 0∼255
　　(&H0∼&HFF)이다.
　　○ 이 명령은 메모리의 내용을 바꿔 놓으므로, 주의하지 않으면
　　에러의 원인이 될 수 있으므로, 사용할 때는 메모리맵을 참조하여,
　　사용할 수 있는 영역인가를 확인해야 한다.

POKE &HD000, 57　;메모리의 &HD000번지에 십진수 57을 써넣는다.

● 참　조 : CLEAR, PEEK, 자료편의 「메모리맵」

POS (포지션)

커서의 수평위치를 얻는다.

● 문형식 : POS(〈식〉)

● 설　명 : ○ 〈식〉은 더미(DUMMY)의 인수이고, 어느것을 지정해도 상관이 없다.
　　보통 0을 사용한다.
　　○ POS는 텍스트 화면에서만 유효하다.

A=POS(0)　;커서의 수평위치를 A에 대입한다.

● 참　조 : CSRLIN

PRESET (피리세트)

● 설　명 : PSET을 참조하십시오.

PRINT (프린트)

문자열이나 수치를 표시한다.

● 문형식 : PRINT〔〈식〉…〕

● 설 명 : ○ 수치를 출력할 때에는 그뒤에 한 문자분의 공백이 확보된다.
○ 수치 앞에는 부호(±)를 표시하기 위한 한칸의 공백이 생기는데 양수(+)이면 생략할 수 있다.
○ 여러개의 문자열, 수치, 변수는 쉼표(,) 또는 세미콜론(;)으로 구분한다.
○ 식의 값이나 문자열을 표시하는 영역은 미리 각 행에 14문자마다 분할되어 정해져 있고, 구분기호로 쉼표를 사용하면 다음 영역의 시작부터 표시된다.
○ PRINT문의 맨 마지막에 세미콜론을 붙이면 행을 바꾸지 않고 다음 PRINT문의 내용을 같은 행에 출력한다.
○ 변수를 포함한 PRINT문에서는 PRINT문 앞에서 변수의 값을 정의하지 않으면, 수치변수의 경우에는 0이 표시되고 문자 변수의 경우는 널스트링이 되므로, 화면에는 아무것도 표시되지 않고 단지 줄만 바뀌게 된다.
○ PRINT 사용방법

PRINT "문자열"	:「" "」로 둘러싸인 문자열을 화면에 출력한다.
PRINT 수치	:수치를 그대로 출력한다.
PRINT 변수	:변수의 값을 출력한다.
PRINT 식	:식의 연산 결과를 출력한다.
PRINT	:한 행분의 공백을 출력(행바꿈을 한다)한다.

○ PRINT는 간략형으로 의문부호(?)를 사용할 수 있다.

```
10 A$="ABC":A=123 ;문자 변수 A$과 수치변수 A의 값을 설정한다.
20 PRINT A$, A    ;A$과 A사이에 쉼표(,)가 있으므로, 데이타의 간격을 13칸 띄워 놓는다.
30 PRINT A$;A     ;A$과 A사이에 세미콜론(;)이 있으므로, 데이타가 연결되어 표시된다.
40 PRINT          ;PRINT만을 지정하면 행바꿈을 해서 한행을 내린다.
50 PRINT 456;"DE"; ;세미콜론에 의해 수치와 문자를 표시한 후 계속 연이어 표시할수 있도록
                   한다.
60 PRINT "GG"     ;행을 바꾸지 않고 50행과 같은행에 표시한다.
```

PRINT # (프린트 넘버)

데이타를 써넣는다.

● **문형식** : PRINT #〈화일번호〉〔,〈식〉…〕

● **설　명** : ○〈화일번호〉는 OPEN문에서 지정한 번호이고,〈식〉은 화일에 써낼 수치 또는 문자열이다.

○ 여러개의〈식〉을 지정할 때는 쉼표(,)나 세미콜론(;)으로 구분한다.

○ 여러개의 문자열을 출력할 수 있지만 INPUT # 이나 LINE INPUT #문으로 읽을 경우 하나의 문자열에는 하나의 변수만이 읽혀진다.

○ OPEN문에서 OUTPUT 모드를 지정하여 화일을 연 다음, 식으로 표시하는 수치나 문자열을 화일에 써낸다.

○「CAS:」(데이타 레코더)에 써넣는 데이타는 INPUT #문, LINE INPUT #문, INPUT$ 함수를 사용해서 읽어 넣을 수 있다.

○ PRINT #문의 끝에 쉼표(,)나 세미콜론(;)이 지정되어 있지 않았을 경우에는 데이타가 출력된 후 데이타의 구분코드로 CR (캐리지리턴) 코드와 LF (라인피드) 코드가 출력되어 진다.

```
10 OPEN "CRT:" FOR OUTPUT AS #1;화일 #1을 출력모드로 하여 CRT화면에 할당한다.
20 A=10:B=20                    ;A, B에 값을 설정한다.
30 PRINT #1, A, B               ;화일 #1의 A, B에 값을 출력한다.
40 CLOSE                        ;화일을 닫는다.
```

● **참　조** : CLOSE, INPUT #, PRINT # USING, LINE INPUT #

PRINT USING (프린트 유징)

문자열이나 수치를 화면에 표시한다.

- **문형식** : PRINT USING 〈서식〉; 〈식〉…
- **설　명** : ○ 〈식〉으로 표시하는 수치나 문자열을 〈서식〉에 따라 텍스트화면에 표시한다.
  ○ 〈식〉은 쉼표(,)나 세미콜론(;)으로 나누어 여러개를 지정할 수가 있는데 왼쪽에 있는 식부터 차례대로 표시된다.
  ○ 다음은 문자열과 수치에서 사용되는 〈서식〉의 기호를 설명한다.

문자열을 편집하는 서식기호

!　　: 문자열의 왼쪽 첫번째 한 문자 만을 표시한다.
& 〈n개의 공백〉 & : 문자열의 첫문자부터 (n+2) 문자를 표시한다.
@　　: 서식의 문자열 안에 포함된 「@」를 식의 문자열로 바꾸어 놓는다.

수치를 편집하는 서식기호

\#　　: 수치를 「#」으로 지정하는 자리수만큼 표시한다. 표시하려고 하는 수치가 〈서식〉에서 지정한 자리수보다 클 경우 수치의 바로앞에 「%」가 표시된다.
.　　: 소숫점의 위치를 지정한다.
\+　　: 〈서식〉의 왼쪽에 「+」를 붙이면, 수식의 앞에 「+, −」의 부호가 표시되고, 〈서식〉의 오른쪽에 「+」를 붙이면, 수식의 뒤에 「+, −」의 부호가 표시된다. 「+」를 2개이상 나열하는 경우 수치 바로 앞의 「+」만이 부호로서의 의미를 갖는다.
−　　: 〈서식〉의 오른쪽에 「−」를 붙이면, 수치가 「−」인 경우에는 수치 다음에 「−」가 표시되고, 서식의 왼쪽에 붙인것은 부호의 의미를 갖지 않는다. 또 오른쪽에 2개이상 나열했을 경우 한개만 부호의 의미를 갖는다.
**　　: 〈서식〉의 왼쪽을 「##」대신에 「**」으로 해놓으면, 수치의 정수부의 자리수가 〈서식〉으로 지정된 자리보다 작을 경우, 왼쪽에 「*」가 붙여진다.
₩₩　　: 〈서식〉의 왼쪽에 「₩₩」를 붙이면, 수치 바로 앞에 「₩」가 표시된다. 지수형식의 서식 지정을 할 경우는 사용되지 않는다.
₩　　: 〈서식〉의 왼쪽에 「₩」을 붙이면, 수치 바로 앞에 「₩」가 표시되고 정수부의 자리수가 〈서식〉으로 지정된 자리수보다 적으면 적은만큼 「*」을 표시한다.
「**₩」는 지수형식의 서식 지정에는 사용하지 않는다.
,　　: 「,」(쉼표)를 정수부의 「#」과 나열하는 경우 수치의 정수부는 3자리마다 「,」로 구분되어 표시되고, 「,」가 소숫점 보다 오른쪽에 놓인 경우는 소숫점 오른쪽에 「,」가 표시되나, 3자리마다 구분하는 것은 행하지 않는다. 「,」는 지수 형식의 〈서식〉을 지정하고 있을 때는 사용할 수 없다.
^^^^　　: 「^^^^」을 자리수 지정의 「#」뒤에 붙이면 수치가 지수형식으로 표시된다. 〈서식〉에 「+」 또는 「−」를 지정하지 않을 경우 공백(수치가 양일때) 또는 「−」를 수치 바로앞에 표시한다.

서식중에 앞에서 설명한 기호 이외의 문자를 붙이면, 문자의 위치에 따라 수치의 앞이나 뒤에 문자가 표시된다. 예제는 아래와 같다.

예) PRINT USING "TAN **###"; 123　;「#」대신 숫자를 써준다.

```
TAN **123
OK
```

4 베이직 설명서

PRINT ＃ USING (프린트 넘버 유징)

문자열이나 수치를 화일에 출력한다.

● 문형식 : PRINT ＃〈화일번호〉, USING〈서식〉;〈식〉…

● 설 명 : ○ 〈식〉으로 표시한 수치나 문자열을 〈서식〉에 따라 지정한 〈화일번호〉의 화일에 출력한다.
○ 출력방법은 〈서식〉을 지정하는 일 외에는 PRINT ＃문과 같다.
○ 〈서식〉을 지정하는 방법은 PRINT USING과 같다.

```
PRINT #1, USING "  ####, ##  —";45;—8  ;화일번호#1에 서식에 따라 수치 45와 —8을
                                        화일에 출력한다.
```

● 참 조 : PRINT ＃, PRINT USING

PSET/PRESET(피세트/피리세트)

점을 찍거나 지운다.

● 문형식 : PSET〔STEP〕(X, Y)〔,〈색번호〉〕〔,〈논리연산기호〉〕
PRESET〔STEP〕(X, Y)〔,〈색번호〉〕〔,〈논리연산기호〉〕

● 설 명 : ○ 좌표(X, Y)의 범위는 SCREEN 모드에 따라 다르다.
○ 〈색번호〉가 생략되면 현재의 칼라색이 지정된다.
○ PRESET 명령에서 색지정을 생략하면 바탕색이 지정되어 자동적으로 점을 지우는 효과를 낸다. 그러나 바탕색 이외의 점을 지정하면 지정한 색의 점이 찍히므로 주의하십시오.
○ 〈논리연산기호〉에 대한 설명은 해설편의「색상」을 참조하십시오.

```
10 SCREEN 3                ;그래픽 모드로 정한다.
20 X=120:Y=96:C=8          ;좌표 및 색깔을 정한다.
30 PSET(X, Y), C           ;좌표(120, 96)에 빨간색의 점을 찍는다.
40 FOR I=0 TO 100:NEXT     ;시간지연
50 PRESET(X, Y)            ;좌표(120, 96)의 점을 지운다.
60 GOTO 60                 ;그래픽 화면을 유지한다.
```

● 참 조 : COLOR, LINE

PUT (풋)

랜덤 화일 버퍼내의 데이타를 랜덤 화일로 출력한다.

●문형식 : PUT〔#〕〈화일번호〉〔,〈레코더번호〉〕

●설　명 : ○〈화일번호〉는 OPEN 명령에서 지정된 것과 같아야 한다.

○〈레코더번호〉를 생략하면 마지막으로 써넣은 다음 레코더가 지정된다.

```
10 OPEN "ET. DAT" AS #1        :「ET. DAT」라는 화일번호 #1을 열어 놓는다.
20 FIELD #1, 3 AS A$           :랜덤화일의 변수(A$)의 자리수를 정한다.
30 GET #1, B$                  :열어놓은 화일로부터 데이타를 읽어낸다.
40 LSET A$=B$:PUT #1, A$:PUT 명령전에 LSET이나 RSET 명령을 사용하여 문자를
                                 랜덤화일 버퍼에 넣는다.
50 CLOSE #1:END                :화일을 닫은후 프로그램을 끝낸다.
```

●참　조 : GET

PUT HAN (풋 한)

SCREEN 5~8에서 2바이트 코드로 정의된 문자를 표시한다.

●문형식 : PUT HAN〔(X, Y)〕,〈코드〉〔,〈색〉〕〔,〈논리연산기호〉〕〔,〈모드〉〕

●설　명 : ○〈코드〉는 MSX_2 표준한글코드 및 자료편의「한자코드(시스템)」를 참조하여 주십시오.

○〈색〉의 지정범위는 각 SCREEN 모드에서 지정할 수 있는 색을 모두 사용할 수 있다.

○〈논리연산기호〉에 대한 설명은 해설편의「색상」을 참조하십시오.

○〈모드〉는 0~2로 지정할 수 있으며 그 의미는 아래와 같다.

0:16×16의 보통크기
1:8×16의 짝수줄
2:8×16의 홀수줄

```
10 SCREEN 6,,,,, 2                    ;화면모드는 홀수페이지와 짝수페이지를 교대로
                                        표시한다.
15 COLOR 3, 0, 0:CLS
20 SET PAGE 0, 0:CLS                  ;0페이지의 화면을 지운다.
30 PUT HAN (128, 90), &HD442,3,, 1;0페이지에 한자 패턴의 짝수줄을 써넣는다.
40 SET PAGE 1, 1:CLS                  ;1페이지의 화면을 지운다.
50 PUT HAN (128, 90), &HD442,3,, 2;1페이지에 한자 패턴의 홀수줄을 써넣는다.
60 GOTO 60
```

위의 프로그램은 화면의 짝수페이지와 홀수페이지에 각각 한자를 표시한후 짝수페이지와 홀수페이지를 교대로 표시한다.

PUT SPRITE (풋 스프라이트)

스프라이트 패턴을 화면에 표시한다.

●**문형식** : PUT SPRITE〈스프라이트면번호〉〔,〔STEP〕(X, Y)〕〔,〈색번호〉〕〔,〈스프라이트 패턴번호〉〕

●**설　명** : ○ 〈스프라이트면번호〉는 0～31로 지정된다.
○ 〈스프라이트패턴번호〉는 SPRITE$ 명령에서 지정한 번호이다.
○ 하나의 스프라이트에는 한개의 패턴밖에 표시할 수 없으며, 이미 표시된 패턴을 지정하면 먼저 표시된 패턴은 지워진다.
○ 스프라이트 패턴이 수평방향으로 5개이상(SCREEN 4～8에서는 9개이상) 나열되면 4개까지(SCREEN 4～8에서는 8개까지)는 표시를 하지만 5번째 (혹은 9번째) 부터는 화면에서 지워진다.
○ 좌표(X, Y)로 패턴을 표시하는 위치를 지정한다. 스프라이트 패턴의 왼쪽 위를 이 좌표로 해서 표시한다.
○ SCREEN 모드에 따라 X는 －32～255(또는 511)의 값을, Y는 －32～191(또는 211)의 값을 지정한다.
○ STEP(X, Y)를 붙이면 최종참조점으로 부터 상대좌표를 구한다. (LINE 명령문 참고)
○ 번호가 서로 다른 패턴이 겹쳐졌을 때 스프라이트 번호가 작은 패턴이 우선적으로 표시된다.
○ 색상은 패턴의 색으로 0～15의 값을 지정한다. 생략하면, 이전의 명령으로 지정된 표시색이 지정된다. 〈색번호〉와 색상에 대해서는 COLOR 명령을 참조하십시오.
○ SCREEN 4이상의 각 스프라이트에서는 수평으로 한라인 마다 색을 지정할 수 있다. 이것에 대해서는 COLOR SPRITE 명령을 참조하십시오.
○ SCREEN 6에서 스프라이트를 사용할 때 SCREEN 5(256×212) 모드를 채택한다. 왜냐하면 SCREEN 6에서는 2도트(DOT)를 하나의 도트로 사용하기 때문에 사실상 512×212의 모드는 256×212로 된다.

PUT SPRITE 1, (98, 45), 8, 1 ;스프라이트 면번호1을 사용하여 좌표(98, 45)에 빨간색으로 표시한다. 그리고 실행이 끝나도 스프라이트는 화면에서 지워지지 않는다.

●**참　조** : COLOR, COLOR SPRITE, SPRITE$, SCREEN

READ (리드)

데이타문에서 지정한 수치나 문자 등의 데이타를 읽어 변수에 대입한다.

●문형식 : READ〈변수〉〔,〈변수〉…〕

●설　명 : ○ 행번호가 작은 DATA 문의 데이타부터 차례대로 읽어 지정한 변수에 대입한다.
그러나 RESTORE 명령을 사용하여 중간의 데이타부터 읽는 경우도 있다.
○ READ 명령에서 지정하는 변수의 형은 데이타형과 같아야 한다.
같지않으면, "Syntax error" (신택스 에러)가 발생되고, 베이직 명령대기 상태로 돌아간다.
○ READ 문의 변수의 갯수가 데이타 갯수보다 많으면,
"Out of DATA" (아웃 오브 데이타) 에러가 발생한다.
○ READ 문의 변수의 갯수가 DATA 문의 데이타 갯수보다 적으면 그다음의 READ 문은
먼저번의 READ 문이 읽었던 데이타의 다음부터 읽기 시작한다.
○ 처음부터 다시 읽고 싶으면 RESTORE 문을 사용한다.

```
10 READ A$, B     ;30행의 데이타를 READ 명령으로 읽는다.
20 PRINT A$, B    ;읽어들인 데이타를 화면에 출력한다.
30 DATA ABC, 1    ;READ 명령에서 읽혀질 데이타들이다.
```

●참　조 : RESTORE, DATA

REM (리마아크)

프로그램에 주석을 넣는다.

●문형식 : REM〔〈주석〉〕

●설　명 : ○ REM 문은 프로그램의 내용을 쉽게 알기위해 사용하며 REM 문은 비실행 문으로
프로그램에 전혀 영향을 주지 않는다.
○ REM 대신에 홑따옴표(')를 사용할 수 있다. 그러나 DATA 문속에서는
REM 문을 사용할 수 없다.

```
10 REM MSX         ;MSX 라는 주석을 달아 주었다.
      ⋮
90 'DAEWOO MSX;REM 대신 「' (홑 따옴표)」를 사용했다.
```

RENUM (리넘버)

프로그램의 행번호를 지정한 순서대로 다시 배열한다.

●**문형식** : RENUM[〈새로운 행번호〉] [, 〈구 행번호〉] [, 〈증가〉]

●**설 명** : ○ 〈새로운 행번호〉와 〈증가〉를 생략하면 10이 지정된다.

○ 〈구 행번호〉를 지정하지 않으면 그 프로그램의 맨 앞부터 〈새로운 행번호〉로 변한다.
예를들면 다음과 같다.

RENUM ;프로그램 전체의 행번호를 바꿔준다.
새로운 행번호 10으로 시작하여 20, 30,…등으로 변한다.

RENUM 15, 50 ;행번호 50을 새로운 행번호 15로 바꾼후 15, 25, 35,…등으로 변화시킨다.

RENUM 30, , 20; 새로운 행번호 30으로 시작하여 50, 70,…등으로 변화시킨다.

○ RENUM은 GOTO, GOSUB, THEN 등 명령문에서 사용하는 행번호도 새로운 행번호로 바꿔준다.

○ GOTO, GOSUB 등으로 지정한 행번호가 프로그램 속에 없는 경우 RENUM 명령을 실행하면 "Undefined line number" (언디파인드 라인넘버)라는 에러 메세지가 표시되고 그 행번호는 그대로 남는다.

○ RENUM 명령은 프로그램의 순서를 바꿀 수 없다. 만일 이와 같은 방법을 사용하면 "Illegal function call" (일레갈 펑션콜) 에러가 발생한다.

RENUM 50, 10, 20 ;구 행번호 10을 새로운 행번호 50으로 하고 그 다음 부터는 20씩 증가한다.

RESTORE (리스토어)

READ문으로 읽을 DATA문을 지정한다.

●**문형식** : RESTORE[〈행번호〉]

●**설 명** : ○ 〈행번호〉는 READ문으로 읽을 데이타의 최초 위치이다.

○ RESTORE 문은 READ문과 함께 사용된다.

○ 〈행번호〉를 생략하면, 맨 처음 DATA 문의 데이타 부터 읽기 시작한다.

RESTORE 50 ;50행에 있는 데이타를 읽는다.

●**참 조** : READ, DATA

RESUME (리숨)

에러처리 루틴을 끝낸후, 프로그램 실행을 다시 시작한다.

●문형식 : RESUME | [0] / NEXT / 〈행번호〉 |

●설　명 : ○ RESUME 0을 지정하면, 에러가 발생한 문에서 부터 다시 프로그램이 실행된다. 0은 생략할 수 있다.
○ RESUME NEXT를 지정하면, 에러가 발생된 다음 문부터 프로그램이 다시 실행된다.
○ 〈행번호〉를 지정하면, 지정된 행부터 다시 실행된다.
○ 에러처리 루틴 이외의 곳에서 RESUME문을 사용하면 "RESUME Without error" (리숨 윋아웃 에러)가 발생한다.

```
RESUME 900   ;에러처리가 끝나면 행번호 900부터 프로그램을 다시 실행한다.
```

●참　조 : ON ERROR GOTO

RETURN (리턴)

서브루틴에서 빠져 나온다.

●문형식 : RETURN [〈행번호〉]

●설　명 : ○ 〈행번호〉를 지정하면, 서브루틴이나 인터럽트 처리루틴을 실행한후, 지정한 행번호로 돌아간다.
○ 〈행번호〉를 지정하지 않으면, 서브루틴을 부른 GOSUB문의 다음 문으로 돌아간다.
○ 인터럽트 처리루틴에서는 처리를 끝낸후, 인터럽트가 걸릴때 중단되었던 프로그램이 다시 실행된다.
○ 서브루틴 중에 CLEAR문을 실행하면, RETURN문에 의해 돌아갈 행번호가 CLEAR되기 때문에 다시 돌아가지 않는다.
○ GOSUB문에 의해 호출되지 않은 상태에서 이 명령을 실행하면, "RETURN Without GOSUB" (리턴 윋아웃 고서브) 에러가 발생한다.
○ 한 개의 서브루틴내에 여러개의 RETURN문이 있어도 상관없다.

```
10 GOSUB 30      ;서브루틴 30을 부른다.
20 END           ;GOSUB문의 다음행에 있으므로 서브루틴이 끝난후 행번호 20으로 온다.
30 PRINT "MSX"   ;행번호 10에서 서브루틴으로 지정한 곳이다.
40 RETURN        ;서브루틴을 부른 그 다음 행으로 되돌아간다.
```

●참　조 : GOSUB, ON GOSUB

RIGHT$ (라이트 스트링)

문자열의 오른쪽부터 지정한 길이의 문자를 얻는다.

● 문형식 : RIGHT$ (〈문자열〉, 〈수식〉)

● 설 명 : ○ 〈수식〉의 범위는 0～255(&H0～&HFF)이다. 0이면 RIGHT$은 널 스트링(NULL STRING)을 값으로 한다.
○ 〈문자열〉의 문자수 보다 〈수식〉이 크면 〈문자열〉모두를 결과로 얻는다.

```
10 A$="HOME COMPUTER"; 문자열 A$을 정한다.
20 PRINT RIGHT$(A$, 8)      ;A$의 오른쪽부터 8개의 문자를 얻는다.
30 END
```

● 참 조 : LEFT$, MID$

RND (랜덤)

0이상 1미만의 난수를 얻는다.

● 문형식 : RND (〈수식〉)

● 설 명 : ○ 발생하는 난수는 RUN문이 실행될 때마다 0이상 1미만의 같은 계열의 배정도 실수값이 나온다. 그러나 「RUN(-TIME)」을 사용하면 매회 나오는 난수를 변화시킬 수 있다.
○ 발생되는 난수는 〈수식〉 값에 의하여 다음과 같이 다르게 된다.
〈수식〉이 양수이면, 새로운 난수를 얻는다.
〈수식〉이 0이면, 바로 앞에 발생한 난수 값을 얻는다.
〈수식〉이 음수이면, 난수 계열을 초기화 한다.

```
A=RND(1)   ;A에는 0이상 1미만의 난수 값이 대입된다.
```

RSET (라이트셀)

랜덤화일 버퍼에 데이타를 오른쪽부터 써넣는다.

● 문형식 : RSET〈문자형변수〉=〈문자식〉

● 설 명 : 데이타를 화일버퍼의 오른쪽부터 써 넣는것 이외에는 LSET명령과 같다.
자세한 설명은 LSET을 참조하십시오.

```
RSET B$=MKS$(ABC)   ;수치변수의 값을 문자형으로 바꾸어 화일 버퍼의 오른쪽부터 써 넣는다.
```

● 참 조 : LSET

RUN (런) ①

메모리에 있는 프로그램을 실행시킨다.

●문형식 : RUN [〈행번호〉]

●설　명 : ○ 메모리에 있는 프로그램 (LIST 명령으로 화면에 표시되는 것)을 실행시킨다.
○ 〈행번호〉를 지정하면 그 행부터 실행이 시작되고, 지정하지 않으면 맨 처음부터 실행이 시작된다.
○ 프로그램의 실행은, 프로그램 중의 END문이나 CTRL + STOP 등에 의해 끝난다.

RUN　;프로그램을 처음부터 실행시킨다.
RUN 20;행번호 20부터 프로그램을 실행시킨다.

RUN (런) ②

화일을 메모리에 읽어들인후 실행시킨다.

●문형식 : RUN"[〈장치명 :〉] 〈화일명〉" [,R]

●설　명 : ○ 〈화일명〉으로 지정한 화일을 읽어들인후 곧바로 실행시킨다.
○ 이 명령을 실행하면 메모리안에 있던 프로그램은 지워지고, 모든 화일은 닫힌다.
○ 〈장치명 :〉은 디스크 드라이브일 경우 "A :" 또는 "B :", 메모리 디스크일 경우 "MEM :" 또는 "MEM 0 :" 으로 지정한다.
현재 사용중인 디스크 드라이브를 지정할 때는 〈장치명 :〉을 생략할 수 있다.
○ 〈화일명〉은 〈장치명 :〉이 디스크 드라이브를 지정하고 있을 경우, SAVE명령이나 BSAVE명령으로 보존되어 있는 화일의 이름이고, 〈장치명 :〉이 메모리 디스크일 경우, SAVE명령으로 보존되어 있는 화일의 이름이다.
○ 〈장치명 :〉이 디스크 드라이브를 지정하고 있을 경우 ,R을 사용할 수 있다.
,R을 사용할 경우 열려져 있는 모든 화일은 닫히지 않은채 실행이 된다.

RUN "TEST"　: 메모리안에 있는 모든 프로그램을 지우고, 열려져 있는 화일을 닫고난 후, 디스켓에 있는 「TEST」라는 화일을 실행시킨다.
RUN "TEST",R　: 디스켓에 있는 「TEST」라는 화일을 읽어들여 실행시키는데 열려져 있던 화일은 닫히지 않는다.
RUN "MEM:SAMPLE": 메모리 디스크안에 있는, 「SAMPLE」라는이 화일을 읽어들여 실행시킨다.

●참　조 : SAVE, BSAVE, LOAD

SAVE (세이브)

메모리에 있는 프로그램을 외부기억장치에 보존한다.

●문형식 : SAVE"〔〈장치명 : 〉〕〈화일명〉"〔,A〕

●설　명 : ○ 메모리에 있는 프로그램을 〈장치명 : 〉이 지정하는 외부기억장치에 보존한다.
○ ,A를 붙이면 메모리에 있는 프로그램을 아스키(ASCII) 형식으로 보존한다.
단, 데이타 레코더에서는 ,A를 사용할 수 없으며, SAVE 명령만으로도 프로그램이 아스키형식으로 보존된다.
○ 아스키 형식으로 보존된 프로그램은 MERGE 명령을 사용하여 메모리내의 프로그램과 합칠 수 있다.
○ 〈장치명 : 〉에 대해서는 해설편의 「화일」을 참조하여 주십시오.

SAVE "CAS:KIM" ; 메모리에 있는 프로그램(KIM)을 데이타 레코더에 아스키형식으로 보존한다.

SAVE "MEM:KANG",A ; 메모리에 있는 프로그램(KANG)을 메모리디스크에 아스키형식으로 보존한다.

●참　조 : LOAD, MERGE

SCREEN (스크린)

화면이나 스프라이트의 크기 등을 설정한다.

●문형식 : SCREEN〔〈화면모드〉〕〔,〈스프라이트 사이즈〉〕〔,〈키 클릭스위치〉〕〔,〈카세트 보레트〉〕〔,〈프린터 옵션〉〕〔〈,인터레이스 모드〉〕

●설　명 : SCREEN문에서 설정되는 항목들은 다음과 같이 사용된다.

―화면모드 지정―

○ 〈화면모드〉의 범위는 0～9이다. (화면모드에 대해서는 해설편의 「화면모드」를 참조하십시오.)
○ 화면을 지정하여 SCREEN 명령을 실행하면 현재 표시하고 있는 화면은 지워지고 새로 지정한 화면모드로 설정된다.
○ SET SCREEN 명령으로 지정하지 않았을 경우 처음 전원을 켜면 베이직의 명령대기 상태는 SCREEN 1 (32×24) 모드가 된다.

0 : 40×24 (또는 80×24) 문자의 텍스트 모드
1 : 32×24문자의 텍스트 모드
2 : 256×192점의 고해상도 그래픽모드
3 : 64×48블록의 저해상도 그래픽모드
4 : 256×192점의 고해상도 그래픽모드
5 : 256×212점의 비트맵 그래픽모드
6 : 512×212점의 비트맵 그래픽모드
7 : 512×212점의 비트맵 그래픽모드
8 : 256×212점의 비트맵 그래픽모드
9 : 80×24문자의 인터레이스 텍스트 모드

○ SCREEN 0, 1, 9의 텍스트모드에서는 화면편집을 할 수 있고 프로그램 리스트도 표시할 수 있다. SCREEN 2～8에서는 점이나 선을 그릴 수 있고, 스프라이트를 표시할 수 있다. (SCREEN 1에서도 스프라이트를 표시할 수 있다)

○ 각각의 모드에서 사용하는 색은 다음과 같다.

0 : 16색중 2색
1 : 16색중 3색 (VPOKE 명령을 사용하면 16색도 가능)
2～5, 7 : 16색
6 : 4색
8 : 256색
9 : 4색

○ 스프라이트는 SCREEN 4～8에서 1라인마다 색을 줄 수 있으며 수평 선상에는 8개까지 나열하여 표시할 수 있다. (COLOR SPRITE 참조)

—스프라이트의 크기—

○ 화면에 표시할 스프라이트 크기를 정한다. 또 SPRITE$ 명령으로 스프라이트를 만들때는 여기서 지정한 것과 같은 크기로 한다. (SPRITE$ 참조)
○ 최초의 상태는 0으로 설정된다.

0 : 8×8 화소로 스프라이트 작성
1 : 8×8 화소의 스프라이트를 가로, 세로로 각각 2배 확대표시
2 : 16×16 화소로 스프라이트 작성
3 : 16×16 화소의 스프라이트를 가로, 세로로 각각 2배 확대표시

○ 화면모드를 변화시키지 않고 〈스프라이트의 크기〉를 설정할 때는 다음과 같이 설정한다.

SCREEN ,2

○ SCREEN 9에서의 스프라이트 크기는 더미 (DUMMY : 아무 의미가 없는 숫자) 이기 때문에 지정을 해도 아무 의미가없다.

—키 클릭 스위치—

○ 키를 누를때 음을 내고 안냄을 정한다.
○ 최초의 상태는 1로 설정된다.

0 : 음을 내지 않는다.
1 : 음을 낸다.

—카세트 보레트—

○ 프로그램을 카세트 테이프에 세이브할 때의 속도를 지정한다.
○ 최초의 상태는 1로 설정된다.

1 : 1200bps
2 : 2400bps (bps : 데이타 전송 속도의 단위)

—프린터 옵션—

○ 인수는 dummy, 즉 의미가 없다.

4 베이직 설명서

—인터레이스(INTERLACE)모드—

○ 화면에 짝수페이지와 홀수페이지를 교대로 표시할 것인가 아닌가를 정한다. (SET PAGE 명령 참조)
○ 인터레이스 모드의 종류는 0~3이다.
 0: 보통(베이직 명령대기상태일 경우)
 1: 인터레이스
 2: Even/Odd 넌(NON) 인터레이스
 3: Even/Odd 인터레이스
○ 〈인터레이스 모드〉 1은 디스플레이 페이지만을 겹쳐서 화면에 표시한다.
○ Even/Odd 모드(2 또는 3)를 지정할 경우, 디스플레이 페이지가 반드시 홀수이어야 하며, 디스플레이 페이지 보다 하나 작은 번호의 페이지와 겹쳐서 화면에 표시한다.
○ SCREEN 9에서 〈인터레이스 모드〉의 값은 더미(DUMMY)이기 때문에 지정을 해도 아무 의미가 없다.

SCREEN,, 1, 2, 0, 1; 키를 누르면 음을 내고, 카세트에 세이브하는 속도를 2400bps로 하고, 프린터는 MSX 표준 프린터를 사용하며 디스플레이 페이지를 겹쳐서 화면에 표시한다.
SCREEN 9,, 0 ; 화면을 80×24문자의 텍스트 모드로 설정하고 키클릭음은 내지 않는다.

●참 조: COLOR, COLOR SPRITE, CSAVE, SPRITE$, SET PAGE

SET ADJUST (세트 어드저스트)

모니터의 화면표시를 상하좌우로 이동한다.

●문형식: SET ADJUST(X, Y)

●설 명: X, Y의 범위는각각 −7~8의 값이다.

SET ADJUST(5, 5) ; 화면 좌상단의 좌표를 우측으로 5화소, 아래로 5화소 이동한다.
SET ADJUST(−4, −3); 화면 좌상단의 좌표를 좌측으로 4화소, 윗쪽으로 3화소 이동한다.

SET BEEP (세트 비프)

BEEP의 음을 설정한다.

●문형식: SET BEEP〈음색〉, 〈크기〉

●설 명: ○ 〈음색〉과 〈크기〉는 각각 1~4의 값이다.
 ○ 〈크기〉의 값이 클수록 음이 크게 울린다.

SET BEEP 1, 1; 크기의 값이 1이므로 음이 거의 나지않는다.
SET BEEP 4, 4; 음색을 변화시킨 음이 울린다.

SET DATE (세트 데이트)

날짜를 설정한다.

●문형식 : SET DATE "〈날짜〉" 〔,A〕

●설　명 : ○ 〈날짜〉는 년, 월, 일을 문자열로 하여 「/」로 구분한다.
　○ ,A를 붙이면 지정한 날짜로 알람(ALARM)이 설정된다.
　○ 년, 월, 일이 한자리인 경우 숫자앞에 0을 붙여 두자리 단위로 지정한다.
　○ 〈날짜〉를 읽고 싶으면 GET DATE 명령을 사용한다.

SET DATE "87/02/19"　;1987년 2월 19일이 설정된다.

●참　조 : GET DATE, SET TIME

SET HAN (세트 한)

한글코드 상태를 변환한다.

●문형식 : SET HAN〔〈문자크기〉〕〔〈,한글코드〉〕〔〈,프린터 옵션〉〕

●설　명 : ○ 현재의 한글코드 체계를 지정한 한글코드 체계상태로 바꾼다.
　○ 〈문자크기〉는 영문자의 크기를 지정하는 것으로, 비트맵 그래픽모드에서만 사용가능하다.
　초기값은 0이고, 다음과 같은 의미를 가진다.

　0: 비트맵 그래픽모드에서 영문자의 크기는 8×8
　1: 비트맵 그래픽모드에서 영문자의 크기는 8×16

　○ 〈한글 코드〉 및 〈프린터 옵션〉은 각각 다음과 같은 한글 코드 상태를 의미한다.

　0: N-바이트 한글 코드 체계 (MSX의 한글 코드와 동일)
　1: 2바이트 분리형 한글 코드 체계 (MS-DOS의 한글 코드와 동일)

　○ 〈한글 코드〉 및 〈프린터 옵션〉을 지정하지 않으면, 이전의 한글 코드 상태를 그대로 유지한다.
　○ 시스템 초기상태에서의 〈한글 코드〉 및 〈프린터 옵션〉은 각각 1, 즉 2바이트 분리형 한글 코드 체계이나, 이전에 SET SYSTEM 명령으로 한글코드가 설정되어 있으면, 그 한글 코드 체계 상태로 된다.
　○ 이 명령을 실행하면 현재의 화면은 지워지고 커서는 좌측상단에 위치한다.
　○ 이 명령을 실행해도 메모리내의 한글 코드는 변하지 않으므로 프로그램을 수정할 때 주의하여 주십시오.

SET HAN, 1, 0　;시스템에서는 2바이트 분리형 한글 코드가 사용되고, 프린터 출력은 N-바이트 한글 코드가 출력된다.
SET HAN 1, 0, 0;영문자의 크기는 8×16이고, 시스템에서는 N-바이트 한글 코드가 사용되며, 프린터 출력도 N-바이트 한글 코드가 출력된다.

●참　조 : SET SYSTEM

SET PAGE (세트 페이지)

화면에 표시할 페이지와 실행할 페이지를 구분한다.

● 문형식 : SET PAGE〈표시 페이지〉, 〈실행 페이지〉

● 설 명 : ○ 〈표시페이지〉는 실제로 모니터에 표시하는 페이지이고, 〈실행페이지〉는 데이타를 써넣는 페이지이다.
○ X-II 는 비디오 메모리가 128KB이나 각 SCREEN마다 실제로 사용되는 메모리는 이것보다 적으므로 남은 용량을 여러개의 페이지로 나누어 사용한다.
○ 베이직명령 대기상태에서의 초기치는 0, 0이다.
○ 이 명령은 화면모드가 5～8일때만 사용 가능하다.

화면모드	페이지 번호
5	0, 1, 2, 3 페이지
6	0, 1, 2, 3 페이지
7	0, 1 페이지
8	0, 1 페이지

○ 페이지를 변환해도 그 페이지의 내용은 초기화가 되지않으므로 페이지의 내용을 모두 지우려면 CLS명령을 실행해야 한다.

SET PAGE 2, 3 ;실행할 페이지는 3이고, 현재 화면에 나타나는 페이지는 2이다.

SET PASSWORD (세트 패스워드)

암호를 설정한다.

● 문형식 : SET PASSWORD "〈암호〉"

● 설 명 : ○ 〈암호〉는 255문자 이내이다.
○ 설정한 〈암호〉를 잊어버린 경우는 [GRAPH] + [STOP] 을 누르면서 전원을 켠다.
○ TITLE 또는 PROMPT를 지정하면 세트시킨 암호는 지워진다.
○ PROMPT, TITLE, PASSWORD를 동시에 지정했을 경우, 맨 마지막에 지정한 명령만 유효하고 나머지는 무시된다.

SET PASSWORD "LEE" ;「LEE」라는 암호를 설정한다.

SET PROMPT (세트 프롬프트)

베이직 프롬프트를 변경한다.

● 문형식 : SET PROMPT "〈프롬프트〉"

● 설 명 : 〈프롬프트〉는 6문자 이내이다.

SET PROMPT "MSX" ;프롬프트 「Ok」대신 「MSX」가 설정된다.

SET SCREEN (세트 스크린)

현재 화면모드의 각 파라메터를 초기치로 설정한다.

● 설 명 : ○ 이 명령을 실행하면 처음 전원을 켤때 SET SCREEN 문으로 정의된 화면의 각 파라메터가 설정된다.
○ 이 명령으로 설정되는 파라메터는 다음과 같다.
• SET ADJUST 로 지정한 X, Y 좌표의 옵셀
• 화면모드, WIDTH, 표시색, 바탕색, 주변색
• 키클릭스위치, 카세트보레트, 프린터 옵션

```
10 SCREEN 0: WIDTH 20;화면을 스크린모드 0으로 하고, 한행을 20문자 길이로 한다.
20 SET SCREEN        ;현재 화면상태를 초기치로 설정한다.
```

SET SYSTEM (세트 시스템)

한글코드 체계의 초기치를 설정한다.

● 문형식 : SET SYSTEM〔〈더미〉〕〔〈, 한글 코드〉〕〔〈, 프린터 옵션〉〕

● 설 명 : ○ 〈더미〉는 아무런 의미가 없다는 뜻이며 지정을 해도 아무런 동작이 수행되지 않습니다.
○ 〈한글코드〉 및 〈프린터 옵션〉에 대해서는 SET HAN 명령을 참조하십시오.

● 참 조 : SET HAN

SET TIME (세트 타임)

시각을 설정한다.

● 문형식 : SET TIME "〈시각〉" 〔, A〕

● 설 명 : ○ 시, 분, 초를 콜론(:) 으로 구분한다.
○ ,A를 붙이면 설정된 시각이 알람(ALARM)으로 설정된다.
○ 시, 분, 초가 한자리일 경우 숫자 앞에 0을 붙여 두자리 단위로 지정한다.
○ 시각을 알고 싶으면 GET TIME 명령으로 읽을 수 있다.

```
SET TIME "11:20:05"   ;11시 20분 5초가 현재의 시각으로 설정된다.
```

● 참 조 : SET DATE, GET TIME

SET TITLE (세트 타이틀)

타이틀을 지정한다.

● 문형식 : SET TITLE 〔"〈타이틀〉"〕

● 설 명 : ○ 〈타이틀〉은 초기화면에 표시되는 것으로 6문자 이내여야 한다.
○ 〈타이틀〉이 6문자 이내이면 다음 화면으로 넘어가고, 6문자로 입력됐을 경우는 아무 키나 누를때까지 초기화면에 머무르고 있다.

> [주의] PASSWORD, TITLE, PROMPT의 기능은 함께 사용 못한다.
> 항상 마지막에 지정된 명령만이 유효하다.

```
SET TITLE "IQ"   ;초기화면에 표시되는 타이틀은 「IQ」라고 설정된다.
```

SET VIDEO (세트 비디오)

수퍼임포즈(SUPER IMPOSE) 모드를 설정한다.

●문형식 : SET VIDEO 〈모드〉 〔, 〈휘도〉 〔, 〈칼라버스〉 〔, 〈동기〉
〔, 〈음성〉 〔, 〈비디오 입력〉 〔, 〈AV 컨트롤〉〕〕〕〕〕〕

●설　명 : ○ 〈모드〉는 수퍼임포즈의 모드를 설정하는 것으로, 다음과 같이 정의된다.

모드	S1	S0	TP	표시화면
0	0	0	0	컴퓨터
1	0	1	1	컴퓨터
2	0	1	0	수퍼임포즈
3	1	0	0	TV

주의 〈모드〉가 0인 때는 외부동기를 사용할 수 없읍니다.
그 밖의 〈모드〉에서는 비디오 프로세스(VDP)의 컴포지트 출력은 사용할 수 없읍니다. S1, S0,TP는 비디오 프로세스의 레지스터 내의 플러그 명칭입니다.

○ 〈휘도〉는 TV 화면 밝기를 제어한다. 값이 0이면 보통상태, 1이면 반으로 된다.
○ 〈칼라 버스〉가 0이면 비디오 프로세스(VDP)의 칼라 버스를 출력모드로 하고, 1이면 입력모드로 한다.
○ 〈동기〉가 1인 때는 외부동기가, 0인때는 내부동기가 선택된다.
○ 〈음성〉은 외부신호를 혼합하여 출력할 것인지 아닌지를 지정하는 것으로, 다음과 같이 정의된다.

음성	기　능
0	외부 음성 신호를 혼합하지 않음.
1	오른쪽 채널의 외부 음성 신호를 혼합함.
2	왼쪽 채널의 외부 음성 신호를 혼합함.
3	양쪽 채널의 외부 음성 신호를 혼합함.

○ 〈비디오 입력〉은 외부 영상 신호의 입력을 선택하는 기능으로, 0이면 RGB 신호, 1이면 외부 영상 신호가 선택된다.
○ 〈AV 컨트롤〉은 RGB 멀티 코넥타의 AV 컨트롤 단자의 출력 여부를 지정하기 위한 것으로, 0이면 OFF, 1이면 ON으로 된다.

주의 수퍼임포즈 기능이 없는 컴퓨터에서는 이 명령을 사용할 수 없읍니다.

SGN (사인)

수치의 부호를 조사한다.

- 문형식 : SGN (〈수식〉)
- 설　명 : 〈수식〉이 양수이면 1, 0이면 0, 음수이면 −1의 값을 얻는다.

A=SGN(20)　;〈수식〉이 양수이므로, A에는 1이 대입된다.
B=SGN(−2.7);〈수식〉이 음수이므로, B에는 −1이 대입된다.

SIN (사인)

삼각함수의 사인값을 얻는다.

- 문형식 : SIN (〈수식〉)
- 설　명 : ○ 〈수식〉의 단위는 라디안이다.
  ○ 결과는 배정도 실수형으로서 −1～1의 값이다.

> 참고 360°는 2π라디안이므로 1도(°)는 2π/360 라디안이고 1라디안은 360/2π도(°)이다.

A=SIN(30*2*3.14159/360)　;30×2×3.14159/360라디안은 30도이므로, A에는 사인 30°의 값인 0.49999961698727이 대입된다.

- 참　조 : COS, TAN

SOUND (사운드)

PSG(음향 발생기)를 직접 작동하여 소리를 낸다.

- 문형식 : SOUND〈레지스터 번호〉, 〈데이타〉
- 설　명 : ○ 〈레지스터 번호〉는 0～13을 지정할 수 있다.
  ○ 〈레지스터 번호〉에 대응하여 각각의 기능을 가지고 있으므로 PLAY문과 같이 간단히 음을 낼수는 없지만 세밀한 지정을 하여 여러가지 음을 낼 수 있다.
  ○ 〈데이타〉는 레지스터 번호에 출력할 정보이며, 각 레지스터에 1개씩 지정된다.
  ○ 음의 발생장치와 음량의 컨트롤장치는 3개의 채널에 독립해 있으므로 따로 지정할 수 있으나, 노이즈(NOISE)의 발생장치와 엔벨로프(ENVELOPE)의 발생장치는 각 채널에서 같이 사용한다.
  ○ PSG의 레지스터는 0～15의 16개가 있고, 각 레지스터의 내용은 다음과 같다.

R0, R1　: 채널 A의 주파수를 지정한다.
R2, R3　: 채널 B의 주파수를 지정한다.
R4, R5　: 채널 C의 주파수를 지정한다.
R6　: 각 채널에서 잡음의 크기를 정한다. (L0～L3을 0으로 하면 음이 안나오고, 15로 하면 최대음량이 나온다.)
R8, R9, R10 : 채널 A, B, C의 각각의 음량을 지정한다.
R11, R12　: 음량 변화주기를 지정한다.
R13, R15　: 엔벨로프의 주기 및 패턴을 지정한다.

—주파수 설정—

○ 각 채널의 주파수 fr은 다음과 같은 식으로 구한다.

$$TP = \frac{fclock}{16 \times fr}$$

fclock : 1.78977MH_Z (1.78977×10^6 Hz) (=R0)
fr　　: 주파수(=R1)

예) 440Hz의 주파수를 출력한다.

$$TP = \frac{1.78977 \times 10^6}{16 \times 440}$$

TP = 약254 (=R0)
fr = 0　　(=R1)

이 값을 256으로 나누어진 몫은 레지스터 1, 3, 5의 값이고, 레지스터 0, 2, 4에 대입되는 값은 나머지가 된다.

—노이즈(잡음) 주파수의 설정—

○ 노이즈 주파수 fn은 다음과 같은 식으로 구한다.

$$NP = \frac{fclock}{16 \times fn}$$

fclock : 1.78977MHz(1.78977×10^6 Hz)
fn　　: 노이즈 주파수(Hz)
NP　　: 레지스터 6에 지정할 값

예) 30Hz의 노이즈를 출력한다.

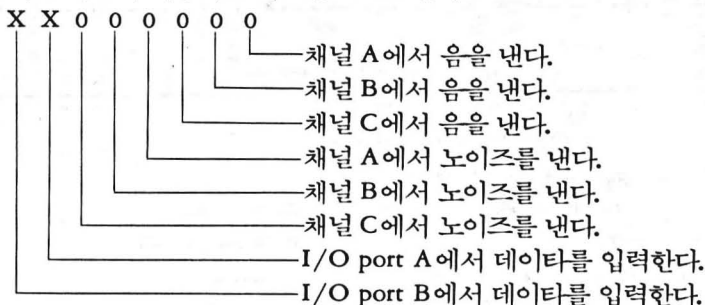
$$NP = \frac{1.78977 \times 10^6}{16 \times (30 \times 10^3)}$$

NP = 약4

레지스터 6에 입력할 값은 4이며, 이것을 SOUND로 지정할 때는 아래와 같이 한다.
(레지스터 6은 3개의 채널에서 공통으로 사용가능하다.)

SOUND 6, 4

—채널의 설정—

○ 어느 채널에서 음 또는 노이즈를 내보낼것 인가를 지정한다.
○ 사용하는 채널에 대응하는 비트를 0으로 한다.

예) SOUND 7, &H111100 ; 채널 A, B에서 음을 내라고 지정한 것이다.

○ 레지스터 7로 채널을 지정할 경우 위와같이 2진수로 지정한 방법이 간단하고 이해하기 쉽다.

—음량 설정—

○ 음의 크기와 음량을 파형에 대응하여 변화시킬 것인가를 지정한다.
○ 음량은 0～15의 값으로 지정하며, 15이면 최대음량이 되고, 0이면 음을 내지 않는다.
○ 음량을 파형으로 변화시킬 때는 16을 지정한다.

—파형 주기의 설정—

○ 파형의 주기 Ep는 다음과 같은 식으로 지정한다.

$$Ep = \frac{fclock \times TE}{256}$$

fclock : 1.78977MHz(1.78977×10^6)
TE : 음이 증가, 감소하는 주기 (초)
Ep : 주기

예) 음이 감소하여 0으로 될 때 까지의 시간을 2초로 한다.

$$Ep = \frac{1.78977 \times 10^6 \times 2}{256}$$

EP = 약 14000
이 값은 256으로 묶은 레지스터 12의 값이고, 레지스터 11의 값은 나머지가 된다.

○ 엔벨로프 발생기의 출력

레지스터 값				엔벨로프 형상
E3	E2	E1	E0	E3-E0
0	0	×	×	
0	1	×	×	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	
→				EP ← 엔벨로프 주기

〈그림 1〉

○ 파형의 주파수는 3개의 채널에서 공통의 레지스터로 지정한다.

—파형의 형태 설정—

○ 파형의 형태에는 〈그림 1〉과 같은 것이 있는데 이중에서 선택된 값을
다음과 같이 지정을 해준다.
예) SOUND 13, 12 ;레지스터 13에 12의 형상을 출력한다.

SOUND 7, 254:SOUND 8, 8 ;고정된 음으로 계속 울린다.

SPACE$ (스페이스 스트링)

지정한 길이의 공백 문자열을 얻는다.

●**문형식** : SPACE$(〈수식〉)

●**설　명** : ○ 〈수식〉의 범위는 0～255이다.
○ 〈수식〉의 수만큼의 공백을 가진 문자열을 얻는다.

```
PRINT "MSX"+SPACE$(1)+"GOOD"
MSX GOOD
```
;MSX 뒤에 한칸의 공백을 준후 GOOD을 표시한다.

●**참　조** : SPC

SPC (스페이스)

화면이나 프린터에 지정한 수만큼 공백을 출력한다.

●**문형식** : SPC(〈수식〉)

●**설　명** : ○ SPC는 PRINT문이나 LPRINT문 등 출력문 중에서만 사용할 수 있고, 지정된 위치에서 부터 〈수식〉만큼의 공백을 표시한다.
○ 〈수식〉의 범위는 0～255이다.

```
PRINT SPC(10);"*"
```
;10칸의 공백을 준후 「*」을 표시한다.

●**참　조** : SPACE$, TAB

SPRITE$ (스프라이트 스트링)

스프라이트 패턴을 만든다.

●**문형식** : SPRITE$(〈스프라이트 패턴번호〉)

●**설　명** : ○ 스프라이트 패턴에 패턴번호를 정의한다.
○ 〈스프라이트 패턴번호〉의 범위는 SCREEN문에서 스프라이트 크기를 0 또는 1 (8×8화소)로 지정하면 0～255이고, 2 또는 3 (16×16화소)으로 지정하면 0～63이다.
○ 스프라이트 패턴을 정의할 때 2진수 형식으로 데이타를 사용하면 이해하기 쉽고 간단하다. (패턴 정의는 1바이트씩)
○ 16×16화소로 스프라이트 패턴을 만들 경우 데이타를 입력하는 순서는 왼쪽의 8개 (1바이트)를 위에서부터 순서대로 읽은다음, 아래쪽 8개를 윗쪽과 같은 방식으로 읽고 오른쪽의 위, 아래 순으로 입력한다.
○ 스프라이트 패턴의 데이타는 8×8화소의 경우 8문자분의 문자열(8바이트), 16×16화소의 경우 32문자분의 문자열(32바이트)이다.
○ 각 1바이트를 한 비트씩 0과 1로 사용하여 1이면 화면에 표시를 하고 0이면 화면에 표시하지 않는다.
○ SCREEN 모드가 4～8인 경우에는 스프라이트의 1라인(LINE)마다 색을 지정하는 것이 가능하다. 이것에 대해서는 COLOR SPRITE 명령을 참조하여 주십시오.

```
10 SCREEN 1, 0                          ;스프라이트 크기를 8×8로 한다.
20 FOR I =1 TO 8                        ;8회 반복을 한다.
30 READ A$:B$=B$+CHR$(VAL("&B"+A$));스프라이트 데이타를 읽어 2진수로 변환하여
                                         문자열로 만든다.
40 NEXT I
50 SPRITE$ (0)=B$                       ;만들어진 스프라이트를 패턴번호 0으로
                                         정한다.
60 PUT SPRITE 0, (128, 96), 3, 0        ;스프라이트 패턴번호 0을 색번호 3으로 하여
                                         좌표(128, 96)에 표시한다.
70  DATA 00010000, 00111000 ┐
80  DATA 01111100, 11111110 │
90  DATA 11111110, 00010000 ├ 스프라이트 데이타
100 DATA 00010000, 00010000 ┘
```

●참　조 : COLOR SPRITE, PUT SPRITE, SCREEN

SPRITE ON/OFF/STOP (스프라이트 온/오프/스톱)

스프라이트 패턴이 겹칠때에 인터럽트를 허가, 금지 또는 보류한다.

●문형식 : SPRITE ON
SPRITE OFF
SPRITE STOP

●설　명 : ○ SPRITE ON 문을 실행하면, PUT SPRITE 문으로 화면에 스프라이트 패턴을 표시할 때, 다른 스프라이트 패턴과 겹쳐지면 인터럽트가 발생하여, 실행중의 프로그램을 중지하고 ON SPRITE GOSUB 문에서 정의한 인터럽트 처리루틴을 실행한다.

○ SPRITE OFF 문은 스프라이트 인터럽트를 금지한다. 이 명령을 실행하면 PUT SPRITE 문으로 화면에 스프라이트 패턴을 출력할때, 다른 스프라이트 패턴과 겹쳐져도 인터럽트가 생기지 않는다.

○ SPRITE STOP 문은 스프라이트의 인터럽트를 보류한다. 이 명령을 실행하면, 스프라이트의 충돌이 일어날 때에도 인터럽트 처리루틴을 실행할 수 없고, SPRITE ON 문을 실행하면, 즉시 인터럽트 처리루틴이 실행된다.

○ SCREEN 4～8의 SPRITE에서 COLOR SPRITE 명령에 따라 특수한 색번호를 지정해 놓으면 그 부분은 인터럽트를 일으키지 않도록 할 수 있다.
이 방법에 대해서는 COLOR SPRITE 명령을 참조하십시오.

●참　조 : COLOR SPRITE, ON SPRITE GOSUB

SQR (스퀘어)

수식의 평방근을 얻는다.

● 문형식 : SQR (〈수식〉)

● 설 명 : ○ 〈수식〉의 값은 0이거나 0보다 커야 한다.
○ 얻어진 값은 배정도형 실수이다.

A=SQR(2) ;A에는 2의 평방근인 1.414213562373이 대입된다.

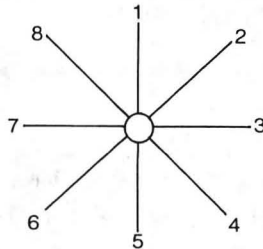
STICK (스틱)

조이스틱이 어느 방향으로 눌러져 있는가를 조사한다.

● 문형식 : STICK (〈조이스틱 번호〉)

● 설 명 : ○ 〈조이스틱 번호〉는 다음과 같이 지정한다.

0: 조이스틱대신 키보드를 사용한다.
1: 포트(PORT)1에 연결되는 조이스틱
2: 포트(PORT)2에 연결되는 조이스틱

○ 조이스틱이 어느 방향으로 눌러져 있는지를 조사하고, 그 결과는 다음 그림에 표시된 값으로 얻는다. 또 조이스틱을 누르지 않으면 값은 0이 된다.

A =STICK(0) ;커서키가 눌러지면 그에 대응한 값이 A에 대입된다.

STOP (스톱)

프로그램의 실행을 정지한다.

● 설 명 : ○ STOP문은 프로그램의 어느곳에, 몇 개를 넣어도 관계없다.
○ STOP문이 실행되어 프로그램이 멈추면 정지한 행을 나타내는 메세지 ("Break in 행번호")가 표시되고 베이직 명령 대기상태로 돌아간다.
○ 정지한 프로그램은 CONT명령을 쓰면 프로그램의 실행을 다시 시작하며, 또 정지한 프로그램을 끝내고 싶으면 [CTRL] + [STOP] 을 한다.

10 PRINT "MSX" ;화면에 「MSX」라고 표시한다.
20 STOP ;화면에 글씨를 표시한후 프로그램을 중단한다.

● 참 조 : CONT, END

STOP ON/OFF/STOP (스톱 온/오프/스톱)

[CTRL] + [STOP] 의 인터럽트를 허가, 금지 또는 보류한다.

●문형식 : STOP ON
STOP OFF
STOP STOP

●설 명 : ○ STOP ON문을 실행하면, [CTRL] + [STOP] 을 할 때, ON STOP GOSUB에서 정의한 처리 서브루틴을 실행한다.
○ STOP OFF문을 실행하면, [CTRL] + [STOP] 을 하여도 정의되어 있는 처리 서브루틴을 실행하지 않는다.
○ STOP STOP을 실행하면 [CTRL] + [STOP] 을 하여도 정의되어 있는 처리 서브루틴을 실행하지 않지만 이후의 프로그램 실행중에 STOP ON을 만나면 즉시 처리 루틴을 실행한다.

●참 조 : ON STOP GOSUB

STR$ (스트링)

수치를 문자열로 바꾼다.

●문형식 : STR$(〈수식〉)

●설 명 : ○ 〈수식〉의 값은 10진수 표기의 문자열로 변환된다.
○ 정수형 또는 실수형 (배정도, 단정도) 의 어떠한 수치도 사용 가능하다.

A$=STR$(80) ;A$에는 10진수 80이 문자열 80으로 변환되어 대입된다.

●참 조 : VAL, STRING$

STRIG (스틱트리거)

조이스틱의 트리거 버튼이 눌러졌는지를 조사한다.

●문형식 : STRIG (〈조이스틱 번호〉)

●설 명 : ○ 지정한 조이스틱의 버튼 또는 스페이스 키를 조사하여, 눌렀으면 −1, 누르지 않았으면 0의 값을 얻는다.
○ 〈조이스틱 번호〉는 5개가 있는데 다음과 같다.
0: 트리거 버튼 대신에 스페이스 키를 사용한다.
1: 포트 1에 접속된 조이스틱의 트리거 버튼 1
2: 포트 2에 접속된 조이스틱의 트리거 버튼 1
3: 포트 1에 접속된 조이스틱의 트리거 버튼 2
4: 포트 2에 접속된 조이스틱의 트리거 버튼 2
○ 단, 조이스틱에 따라서 트리거 버튼 2가 없는 것도 있다.

A=STRIG(1) ;포트 1에 조이스틱 버튼 1이 눌러지면 A에 −1을 대입한다.

●참 조 : ON STRIG GOSUB, STRIG(n) ON/OFF/STOP

STRIG(n) ON/OFF/STOP (스틱트리거 온/오프/스톱)

조이스틱의 트리거 버튼에 의한 인터럽트를 허가, 금지 또는 보류한다.

● 문형식 : STRIG (<조이스틱 번호>) ON
STRIG (<조이스틱 번호>) OFF
STRIG (<조이스틱 번호>) STOP

● 설 명 : ○ <조이스틱 번호>는 STRIG 명령을 참조하십시오.
○ STRIG ON을 실행하면, 지정된 트리거 버튼이 눌러졌을 때 ON STRIG GOSUB 문으로 정의한 서브루틴을 실행한다.
○ STRIG OFF를 실행하면, 지정된 트리거 버튼을 눌러도, 정의된 서브루틴은 실행되지 않는다.
○ STRIG STOP을 실행하면, 지정된 트리거 버튼이 눌러져도 정의된 서브루틴은 실행되지 않지만, 이후의 프로그램 실행중에서 STRIG ON 명령을 만나면 처리루틴을 실행한다. 이 명령은 STRIG ON의 상태에서만 유효하다.

STRIG(0) ON ;스페이스키가 눌러지면 인터럽트를 허가한다.

● 참 조 : ON STRIG GOSUB, KEY ON/OFF/STOP, STRIG

STRING$ (스트링)

문자를 지정한 수만큼 얻는다.

● 문형식 : STRING$ (<식>, |<문자식> / <수식>|)

● 설 명 : ○ <문자식> 혹은 <수식>으로 지정한 문자를 <식>의 갯수만큼 얻는다.
<문자식>이 문자열일 경우에는, 문자열의 처음 한 문자만을 취한다.
○ <수식>의 범위는 0~255이고, 그 값을 문자코드로 하는 문자를 의미한다.

A$=STRING$(3, "*") ;A$에는 문자 「*」가 3개 대입된다.

● 참 조 : STR$, VAL

SWAP (스와프)

두 변수의 값을 교환한다.

● 문형식 : SWAP <변수>, <변수>

● 설 명 : ○ 서로 교환하는 <변수>의 형은 일치해야 한다. 일치하지 않으면 "Type mismatch" (타입 미스매치) 에러가 발생한다.
○ <변수>의 형은 정수, 단정도, 배정도, 문자 또는 배열변수이다.

SWAP A, B ;A와 B의 값을 서로 바꾸어 갖는다.

TAB (탭)

지정된 자리까지 공백을 출력한다.

●문형식 : TAB (〈수식〉)

●설 명 : ○ TAB 명령은 현재 수평위치의 맨 좌측부터 지정된 수만큼의 공백을 표시한다.
○ 〈수식〉의 범위는 0～255이다.
○ PRINT나 LPRINT의 출력문 중에서 사용한다.
○ 현재의 위치가 지정한 〈수식〉의 수평위치보다 크면 TAB기능이 무시된다.

PRINT TAB(5); "MSX" ;5칸의 공백을 준 후 6번째 자리에 「MSX」가 표시된다.

●참 조 : SPC

TAN (탄젠트)

삼각함수의 탄젠트 값을 얻는다.

●문형식 : TAN (〈수식〉)

●설 명 : ○ 〈수식〉의 단위는 라디안이다.
○ 결과는 배정도 실수형으로 얻어진다.

A=TAN(30 * 2 * 3.14159/360) ;30×2×3.14159/360 라디안은 30° (도) 이므로, A에는 탄젠트 30° 의 값인 0.57734967950317이 대입된다.

●참 조 : SIN, COS

TIME (타임)

1/60초 마다 값을 1씩 증가시켜 시간을 잰다.

●문형식 : TIME〔=〈수식〉〕

●설 명 : ○ 〈수식〉의 범위는 0～65535이다.
○ 프로그램 실행시간을 계산할 때, 처리의 처음 시작에는 TIME=0을 대입하고, 처리의 맨 마지막에 TIME을 조사하면 1/60단위의 시간을 얻는다.
○ 이 명령은 VDP 레지스터가 1/60초마다 인터럽트를 걸 때, 값을 바꾸기 때문에, 데이타 레코더를 사용하고 있는 동안이나, 인터럽트를 걸 수 없는 동안은 타임의 값이 바뀌지 않는다.

```
10 T=TIME:PRINT T    ;경과되는 시간(초)이 화면에 표시된다.
20 IF T=60 THEN BEEP;T값이 60이면 삑 소리를 낸다.
30 END               ;프로그램을 끝낸다.
```

●참 조 : ON INTERVAL GOSUB

4

베이직 설명서

TRON/TROFF (트레이스 온/트레이스 오프)

프로그램의 실행상태를 추적한다.

●설 명 : ○ TRON 명령을 실행하면, 실행되는 프로그램의 행번호가 대괄호([ ])로 묶여서 화면에 표시된다. 이 명령은 프로그램의 디버그(DEBUG)를 위해서 사용되며, 실행은 직접모드 또는 간접모드에서 가능하다.
○ TROFF 명령을 실행하면, 프로그램 추적이 불가능해진다.
○ 텍스트모드에서만 사용할 수 있다.

USR (유저)

기계어로 작성된 루틴을 호출한다.

●문형식 : USR [〈번호〉] (〈인수〉)

●설 명 : ○ 미리 메모리안에 준비해 놓은 기계어 루틴을 호출하며, 실행개시 번지는 DEF USR 명령으로 미리 지정해야 한다.
○ 〈번호〉는 0~9의 값으로 DEF USR 명령으로 정의된 번호에 대응한다. (〈번호〉가 생략된 경우는 0으로 간주된다)
○ 〈인수〉는 BASIC에서 기계어 루틴으로 전해주는 값이다. 〈인수〉는 생략해서는 안되며, 호출하는 루틴에 〈인수〉를 전해줄 필요가 없는 경우에도 더미(DUMMY) 인수를 지정한다.

```
I = USR 3(J)  ;인수를 J로 하는 사용자함수, 3의 계산 결과치를  I에 대입한다.
```

●참 조 : DEF USR

VAL (벨류)

문자열을 수치값으로 얻는다.

●문형식 : VAL (〈문자식〉)

●설 명 : ○ 〈문자식〉으로 표시한 문자열을 수치로 변환하여, 그 결과를 얻는다.
○ 문자열의 처음 문자가 영문자, /, * 등 숫자 이외의 것(단, &B, &H는 제외)이면, VAL 함수의 값은 0이 된다.
○ 문자열 중의 공백은 무시된다.

```
10 A$="80":B$="-45"          ;문자변수 A$과 B$의 값을 지정한다.
20 C$=A$+B$                  ;A$+B$을 C$에 대입한다.
30 A=VAL(A$): B=VAL(B$)      ;A$과 B$을 수치값으로 바꿔 A, B에 대입한다.
40 C=A+B                     ;수치로 바꾼 A, B를 더하여 C에 대입한다.
50 PRINT C$, C:END           ;C$과 C의 결과를 화면에 표시한다.
```

●참 조 : STR$

VARPTR (베리어블 포인터)

변수 또는 메모리에 있는 화일컨트롤 블록의 시작 번지를 얻는다.

●문형식 : VARPTR (〈변수명〉)
VARPTR (#〈화일번호〉)

●설　명 : ○ 〈변수명〉으로 지정한 변수의 데이타가 저장되어 있는 변수 영역의 메모리 번지를 구한다.
○지정된 〈화일번호〉에 할당되어 있는 화일 컨트롤 블록의 개시번지를 얻는다.
○화일 컨트롤 블록이란, 외부장치에 데이타를 쓰거나 읽어들일 경우에 사용되는 버퍼로서 오픈(OPEN) 문을 지정할 경우에 메모리에 할당되어진다.

PRINT HEX$(VARPTR(B)) ; B의 데이타가 저장되어 있는 메모리의 첫번째 바이트의 번지를 얻는다.
X=VARPTR(#1) ; 화일번호 #1에 할당되어 있는 화일 컨트롤 블록의 시작번지를 얻는다.

●참　조 : 자료편의 「메모리맵」

VDP (브이디피)

VDP (Video Processor) 레지스터의 값을 읽거나 바꿔준다.

●문형식 : VDP (〈레지스터 번호〉)

●설　명 : ○ VDP는 파라메터 −1∼−9, 0∼24 및 33∼48에 대응하는 레지스터에 데이타를 써넣거나 읽어낸다.
○ 이 함수는 V9938의 규격을 이해한 후 사용하여 주십시오.

VDP(1)=&HE1 ;VDP 레지스터 1에 「&HE1」라는 데이타를 써넣는다.

●참　조 : BASE

VPEEK (브이피크)

비디오메모리의 지정한 번지의 내용을 읽는다.

●문형식 : VPEEK (〈번지〉)

●설　명 : ○ VPEEK 명령은 비디오 메모리에 기록되어 있는 1바이트(BYTE)분의 정보를 읽어낸다.
○ 지정하는 〈번지〉는 SET PAGE 명령으로 설정한 페이지를 옵셀으로한 0∼65535(&H0∼&HFFFF)의 값이다.
○ 비디오 메모리란 화면에 관계되는 정보를 가지고 있는 메모리로서 화면에 표시되는 점이나 문자의 위치, 색, 형태 등의 데이타가 저장되어 있다.

A=VPEEK(0) ;VRAM 0번지에 있는 값을 변수 A에 대입한다.

●참　조 : VPOKE, BASE

VPOKE (브이포크)

비디오 메모리의 지정한 번지에 데이타를 써넣는다.

● 문형식 : VPOKE 〈번지〉, 〈데이타〉

● 설　명 : ○ VPOKE 명령은, 비디오 메모리에 데이타를 직접 써넣는다.
○ 〈번지〉는 0∼65535의 값을 갖는 수식으로 지정하며, SET PAGE 명령으로 설정한 페이지를 옵셀으로한 값이다.
○ 〈데이타〉는 1바이트(0∼255)의 값을 갖는 수식으로 지정한다.

VPOKE&H58, &HB0　;데이타 &HB0를 &H58번지에 써넣는다.

● 참　조 : SET PAGE, VPEEK

WAIT (웨이트)

입력포트의 상태를 화면을 통해 보는 동안, 프로그램 실행은 잠시 중단된다.

● 문형식 : WAIT 〈포트번호〉, 〈식1〉 [, 〈식2〉]

● 설　명 : ○ 〈포트번호〉로 지정한 입력포트에서 읽어들인 데이타와 〈식2〉를 XOR 한다. 이 결과를 다시 〈식1〉과 AND 한다.
○ 만약 그 결과가 0 (거짓)이면 BASIC은 또 한번 포트의 상태를 읽어들여 같은 동작을 반복한다. 결과가 0이 아니면 (참) 프로그램 상태는 다음문을 실행한다.
○ 〈식2〉가 생략된 경우는 0으로 본다.

WAIT10, &H22, &H22 ; 입력포트10에서 읽어들인 데이타를 &H22와 XOR하고, 그 결과를 다시 &H22와 AND하여 그 결과가 구해질 때까지 프로그램의 실행을 중지한다.

WIDTH (위쓰)

1행에 표시하는 문자수를 정한다.

● 문형식 : WIDTH 〈자리수〉

● 설　명 : ○ 〈자리수〉의 범위는 SCREEN 1의 경우에는 1∼32로 지정할 수 있으며, SCREEN 0와 SCREEN 9에서는 1∼80를 지정할 수 있다.
○ SCREEN 9에서 한자를 표시할 경우, 〈자리수〉의 범위는 0∼64로 해야한다.

WIDTH 80　;텍스트 모드의 화면에서 1행의 문자수를 80문자로 설정하여 표시한다.

● 참　조 : SCREEN, SET SCREEN

5

예제 프로그램

1. 최대 최소값

READ문으로 읽어들인 숫자를 비교하여 최대값과 최소값을 구하는 프로그램입니다.

```
10 REM ** EXAMPLE **
20 DIM ND(20)
30 MX=0
40 MI=99
50 FOR I=1 TO 20
60 READ ND(I)
70 NEXT I
80 FOR I=1 TO 20
90 IF MX>=ND(I) THEN 110
100 MX=ND(I)
110 IF MI<=ND(I) THEN 130
120 MI=ND(I)
130 NEXT I
140 PRINT "MAXIMUM=";MX
150 PRINT "MINIMUM=";MI
160 DATA 22,32,23,11,19,50,61,18,13,31,35
170 DATA 39,98,7,4,44,77,12,82,65
180 END
```

2. 전자펜

전자펜을 사용하여 화면에 그림을 그릴 수 있게 해주는 프로그램입니다.

```
10 'light sample program
20 COLOR 15,4,7:SCREEN2
30 CALL ADJUST
40 IF NOT PAD(20) THEN 40
50 IF NOT PAD(23) THEN 40
60 X=PAD(21):Y=PAD(22)
70 PSET(X,Y),1
80 GOTO 40
```

3. PLAY

2개의 채널을 사용하여 음악을 연주하는 프로그램입니다.

```
10 'Bach
20 PLAY"T240L6V12","T240L2V9"
30 PLAY"R806GAB07DCCED","04G05GE"
40 PLAY"DGF#GD06BGAB","04B05E04E"
50 PLAY"07CDEDC06BGABG","04AB05C"
60 PLAY"F#GADF#A07C06BA","05DF#D"
70 PLAY"BGA807DCCED","GGC"
80 PLAY"DGF#GD06BGAB","04B05ED"
90 PLAY"E07DC06BAGDGF#G2","CC#DG"
100 IF PLAY(0) THEN 100
110 PRINT "*** THE END ***"
```

4. INKEY $

INKEY$를 이용한 프로그램입니다. 커서를 움직여 마음대로 그림을 그릴 수 있습니다.

```
10 SCREEN 2
20 IF INKEY$<>"" THEN 20
30 I$=INKEY$
40 IF I$="" THEN 30
50 A=ASC(I$)
60 IF A<28 OR A>31 THEN 30
70 ON A-27 GOTO 80,90,100,110
80 X=X+1:IF X>255 THEN X=1:GOTO 120 ELSE 120
90 X=X-1:IF X<1 THEN X=255:GOTO 120 ELSE 120
100 Y=Y-1:IF Y<1 THEN Y=191:GOTO 120 ELSE 120
110 Y=Y+1:IF Y>191 THEN Y=1
120 PSET(X,Y)
130 GOTO 20
```

5. 그래픽(I)

LINE 명령을 이용하여 멋진 그래픽을 나타내는 프로그램입니다.

```
10 SCREEN 2
20 FOR I=0 TO 95 STEP 3
30 LINE(128-I,95-I)-(128+I,95+I),7,B
40 NEXT
50 FOR J=0 TO 400:NEXT
60 FOR K=1 TO 15 STEP 7
70 FOR X=K TO 255 STEP 5
80 LINE(X,0)-(255-X,191),7
90 NEXT X
100 FOR Y=191-K TO 0 STEP -5
110 LINE(0,Y)-(255,191-Y),4
120 NEXT Y
130 NEXT K
140 GOTO 140
```

6. 그래픽(II)

CIRCLE 명령을 사용하여 입체감있는 그림을 나타내주는 프로그램입니다.

```
10 COLOR ,1,1:SCREEN2,3:PI=3.14159#
20 XX=84:YY=65
30 FOR TH=0 TO 180 STEP 4
40 T=TH/180*PI
50 YU=ABS(TAN(T))
60 IF YU=0 THEN YU=1E-03
70 TU=1/YU
80 CIRCLE(XX,YY),50,9,,,TU
90 IF TU>1 THEN 110
100 XX=XX+4:YY=YY+2
110 NEXT TH
120 FOR KK=1 TO 1000:NEXT KK
130 LINE(0,0)-(255,191),1,BF
140 GOTO20
```

7. COLOR

SCREEN 8에서 사용할 수 있는 0~255의 색상을 화면에 나타내주는 프로그램입니다.
프로그램의 실행을 끝내려면, [CTRL] + [STOP]을 누르세요.

```
100 ON STOP GOSUB 200:STOP ON
110 SCREEN8:COLOR 0,0,0:CLS
120 K=0:Y=8
130 FOR I=0TO 15
140 X=32:Y=Y+12
150 FOR J=0TO 15
160 LINE(X,Y)-(X+8,Y+8),K,BF
170 X=X+12:K=K+1
180 NEXT J,I
190 GOTO 190
200 STOP OFF
210 COLOR 15,4,7
220 END
```

8. PALETTE(빠레트)

색변화를 나타내기 위한 프로그램입니다.
빠레트는 512색 중에서 16색을 선택하여 사용하기 위한 기능입니다.
모든 색을 사용하여 전체의 색을 출력하여 봅시다.

```
10 '
20 'RGB TEST PROGRAM
30 '
40 SCREEN 0
50 COLOR 15,0
60 CLS
70 LOCATE 15,8:PRINT "R  G  B"
80 FOR R=0 TO 7
90 FOR G=0 TO 7
100 FOR B=0 TO 7
110 LOCATE 14,10:PRINT R;G;B
120 LOCATE 14,12
130 R$=RIGHT$("000"+BIN$(R),3)
140 G$=RIGHT$("000"+BIN$(G),3)
150 B$=RIGHT$("000"+BIN$(B),3)
160 PRINT R$+G$+B$:COLOR =(0,R,G,B)
170 FOR I=0 TO 500:NEXT
180 NEXT B,G,R
190 GOTO 80
```

9. SCREEN 8

이 프로그램은 SCREEN 8의 색을 이용하여 하나의 나비를 그리고, 그것을 축소하여 Copy 명령에 의해 여러개의 나비를 화면에 나타내줍니다.
SCREEN 8에서는 256의 고정색을 동시에 사용할 수 있기 때문에 항상 많은 색을 표현할 수 있읍니다.

```
10 '
100 '
110 'LINE SHRINK COPY BUTTERFLY
120 '
130 DEFINT A-Z:DIM A(1699)
140 DEFFNP=INT(RND(1)*2)+2 OR INT(RND(1)*3+3)*32
150 '*** BUTTERFLY ***
160 SCREEN 8:COLOR 227,0,0:CLS
170 PSET(128,100):FOR J=0 TO 1
180 FOR I=0 TO 14:READ X,Y
190 IF J THEN X=-X:IF I=14 THEN 210
200 LINE-STEP(X*2,Y*2)
210 NEXT:RESTORE:NEXT
220 LINE-(128,100)
230 DATA 30,-30,10,-5,5,0,2,10,0,9
240 DATA -2,15,-3,10,-1,9,-2,6,-3,3
250 DATA -5,2,-12,-2,-11,-7,-8,-11,0,-9
260 PAINT(130,100),227,227
270 PAINT(120,100),227,227
280 FOR X=128 TO 255 STEP 2:P=FNP
290 LINE(128,100)-(X,0),P,,AND
300 LINE(128,100)-(256-X,0),P,,AND
310 NEXT
320 FOR Y=0 TO 211 STEP 2:P=FNP
330 LINE(128,100)-(255,Y),P,,AND
340 LINE(128,100)-(0,Y),P,,AND
350 NEXT
360 FOR X=255 TO 128 STEP -2:P=FNP
370 LINE(128,100)-(X,211),P,,AND
380 LINE(128,100)-(255-X,211),P,,AND
390 NEXT
400 PSET(138,80),227
410 LINE-(128,100)
420 LINE-(128,130)
430 PSET(127,124),227
440 LINE-(127,130)
450 LINE-(127,100)
460 LINE-(118,80)
470 '*** SHRINK ***
480 FOR Y=0 TO 52
490 FOR X=0 TO 63
```

```
500 PSET(X,Y),POINT(X*4,Y*4)
510 NEXT X,Y
520 '*** COPY ***
530 COPY(0,0)-(63,52),0 TO A
540 FOR Y=0 TO 211 STEP 53
550 FOR X=64 TO 255 STEP 128
560 COPY A,1 TO (X-1,Y)
570 COPY A,2 TO (X,Y+52)
580 NEXT X,Y
590 LINE(0,0)-(255,211),255,BF,XOR
600 GOTO 590
```

10. GRADATION(그레데이션)

빠레트에 같은 계열의 색을 사용하여 GRADATION의 효과를 얻어 사선으로 나타내줍니다.

```
10 '
20 ' screen5 palette demo
30 '
40 COLOR 15,0,0:SCREEN 5
50 FOR I=0 TO 7:READ A,B,C,D
60 GOSUB 100:GOSUB 160
70 NEXT:RESTORE
80 GOTO 50
90 '*** set color ***
100 FOR J=0TO7
110 COLOR=(J+A*8,J*B,J*C,J*D)
120 NEXT:RETURN
130 DATA,,,1,1,,1,,,1,,,1,1,1,1
140 DATA,,1,1,1,1,,1,,1,1,,1,,,
150 '*** line ***
160 E=A*8:F=E:N=0:M=1
170 FOR X=0TO 255
180 IF X>211 THEN N=N+1
190 LINE(X,255*A)-(N,ABS(A*211-X+N)),E
200 E=E+M:IF E=8+F THEN E=E-1:M=-M
210 IF E<F THEN E=F:M=-M
220 NEXT
230 FOR Y=209*A+1 TO 211*(1-A) STEP 1-A*2
240 N=N+1:LINE(255,Y)-(N,211*(1-A)),E
250 E=E+M:IF E=8+F THEN E=E-1:M=-M
260 IF E<F THEN E=F:M=-M
270 NEXT:RETURN
```

11. SPRITE(I)

스프라이트를 이용하여 나비를 만든 후 이것을 랜덤하게 나타내는 프로그램입니다.

```
10 DEFINT A-Z
20 DIM X(10),Y(10)
30 SCREEN2,3:COLOR ,1,1:CLS:I=RND(-TIME)
40 FOR I=0 TO 10:X(I)=96:Y(I)=I*15:NEXT
50 FOR I=0 TO 31:READ A$:B$=B$+CHR$(VAL("&h"+A$)):NEXT
60 SPRITE$(0)=B$
70 FOR I=1 TO 10
80 PUT SPRITE I,(X(I),Y(I)),I+4,0
90 NEXT
100 FOR I=1 TO 10
110 X(I)=(X(I)+RND(I)*21-10)MOD 256
120 Y(I)=(Y(I)+RND(I)*21-10)MOD 192
130 NEXT
140 GOTO 70
150 DATA 0c,06,62,f2,fe,dd,cf,c7
160 DATA ff,7f,3f,1b,37,3e,1c,00
170 DATA 30,60,46,4f,5f,bb,f3,e3
180 DATA ff,fe,fc,d8,ec,7c,38,00
```

12. SPRITE(Ⅱ)

SCREEN 4부터 8까지의 스프라이트에서는 수평라인마다 색을 칠할 수 있으며, 칼라코드(색번호)의 비트6을 1로하면, 스프라이트를 간단히 동작시킬 수 있습니다.
이 프로그램을 실행하면, 4개의 스프라이트가 겹쳐서 표시되며, 이때, 아무키나 누르면 차례로 스프라이트가 분리됩니다.
프로그램을 끝내려면, CTRL + STOP 을 누르세요.

```
10 '
20 'color sprite bit6 sample
30 '
40 ON STOP GOSUB 370:STOP ON
50 COLOR 15,15,15:SCREEN 5,3
60 T=85:S$="":FOR I=1 TO 32:GOSUB 340:NEXT
70 SPRITE$(0)=S$
80 T=51:S$="":FOR I=1 TO 32:GOSUB 340:NEXT
90 SPRITE$(1)=S$
100 T=15:S$="":FOR I=1 TO 32:GOSUB340:NEXT
110 SPRITE$(2)=S$
120 T=1:S$=CHR$(0)
130 FOR I=1 TO 7:S$=S$+CHR$(T):T=T*2+1:NEXT
140 FOR I=1 TO 8:S$=S$+CHR$(T):T=(T*2)MOD256:NEXT
150 T$=RIGHT$(S$,8)+LEFT$(S$,8):S$=S$+T$
160 SPRITE$(3)=S$
170 COLOR SPRITE$(0)=STRING$(16,1)
180 COLOR SPRITE$(1)=STRING$(16,66)
190 COLOR SPRITE$(2)=STRING$(16,68)
200 COLOR SPRITE$(3)=STRING$(16,72)
210 COLOR=(0,0,0,0):COLOR=(8,1,1,1)
220 COLOR=(1,0,0,7):COLOR=(9,0,0,4)
230 COLOR=(2,0,7,0):COLOR=(10,0,4,0)
240 COLOR=(3,0,7,7):COLOR=(11,0,4,4)
250 COLOR=(4,7,0,0):COLOR=(12,4,0,0)
260 COLOR=(5,7,0,7):COLOR=(13,4,0,4)
270 COLOR=(6,7,7,0):COLOR=(14,4,4,0)
280 COLOR=(7,7,7,7):COLOR=(15,4,4,4)
290 PUT SPRITE 0,(32,100):A$=INPUT$(1)
300 PUT SPRITE 1,(64,100):A$=INPUT$(1)
310 PUT SPRITE 2,(96,100):A$=INPUT$(1)
320 PUT SPRITE 3,(128,100):A$=INPUT$(1)
330 GOTO 290
340 S$=S$+CHR$(T)
350 T=(T*2)MOD256+T₩128
360 RETURN
370 STOP OFF
380 COLOR 15,4,7
390 END
```

13. TIME

TIME명령을 이용하여 실제로 시계효과를 내주는 프로그램입니다.

```
10 CLS
20 LOCATE 8,8
30 PRINT"start!"
40 TIME=0
50 LOCATE 8,10
60 T=TIME
70 H=INT(T/3600)
80 M=INT(T/60):IF M>59 THEN M=INT(M MOD 60)
90 S=T MOD 60
100 PRINT USING"## : ##'  ##";H;M;S
110 IF H=9 THEN END
120 GOTO 50
```

14. GET DATE, TIME

날짜와 시각을 화면에 나타내주는 프로그램입니다.

```
10 COLOR 15,0,0:SCREEN7
20 OPEN"grp:"FOR OUTPUT AS#1
30 FOR I=1TO300
40 PSET(RND(1)*512,RND(1)*212),RND(1)*15
50 NEXTI
60 X=0:Y=0
70 COLOR =(RND(1)*6+1,RND(1)*7,RND(1)*7,RND(1)*7)
80 LINE(0,Y)-(512,Y),RND(1)*15
90 LINE(0,212-Y)-(512,212-Y),RND(1)*15
100 LINE(X,0)-(X,212),RND(1)*15
110 LINE(512-X,0)-(512-X,212),RND(1)*15
120 GET DATE A$:GET TIME B$
130 PRESET(172,100),0:PRINT#1,"  ";A$;"  ";B$
140 X=X+1:Y=Y+1
150 IF Y=48 THEN 60 ELSE 70
```

15. SET PAGE

VRAM에 128K 바이트가 있기 때문에 각각의 SCREEN에서 어떤 PAGE의 사용도 가능하며, PAGE를 서로 바꿔서 사용할 수도 있읍니다.

```
10 '
20 'PAGE SAMPLE
30 '
40 SCREEN5:COLOR 15,1,1:CLS
50 DEFINT A-Z:DIM A(259)
60 FOR I=0TO3:SET PAGE 0,I:CLS:NEXTI
70 SET PAGE 0,0
80 CIRCLE(48,48),15,15,-1,-3.14
90 PAINT(40,40),8,15
100 COPY(32,32)-(63,63),0 TO A
110 FOR I=0 TO 3:SET PAGE 0,I
120 COPY A,I TO (32,32),I
130 NEXT I
140 FOR I=0TO 3
150 SET PAGE I,I
160 FOR J=0TO 1000:NEXTJ
170 NEXTI
180 GOTO 140
```

16. SET ADJUST

여러개의 원을 일정한 간격으로 화면에 그려, 그 화면을 SET ADJUST 명령에 의해 상하 좌우로 움직이게 하는 프로그램입니다.

```
10 COLOR 15,0,0:SCREEN5
20 FOR I=0 TO 80 STEP 4
30 CIRCLE(128,105),I
40 NEXT
50 DATA 0,-7,1,-7,2,-7,3,-6,4,-6
60 DATA 5,-5,6,-4,6,-3,7,-2,7,-1
70 DATA 7,0,7,1,7,2,6,3,6,4
80 DATA 5,5,4,6,3,6,2,7,1,7
90 DATA 0,7,-1,7,-2,7,-3,6,-4,6
100 DATA -5,5,-6,4,-6,3,-7,2,-7,1
110 DATA -7,0,-7,-1,-7,-2,-6,-3,-6,-4
120 DATA -5,-5,-4,-6,-3,-6,-2,-7,-1,-7
130 FOR I=1 TO 40
140 READ X,Y
150 SET ADJUST(X,Y)
160 NEXT:RESTORE
170 GOTO 130
```

17. SET VIDEO

외부비디오 신호와 컴퓨터의 신호를 합성하는 예를 보여주는 프로그램입니다. 실행시키기 전에 각 부분의 연결상태를 다시한번 확인해 주십시오.

```
10 SCREEN8:SET HAN1,1,1:SET PAGE 0,0:COLOR 255,0,0:CLS
20 SET VIDEO 2,0,0,1,3,1,1
30 OPEN"grp:" FOR OUTPUT AS#1
40 READ X,Y,C1,C2,A$:IF A$="##" THEN 160
50 COLOR C1
60 PRESET(X*16,Y*16),,PSET:PRINT#1,A$
70 PRESET(X*16+1,Y*16),,TOR:PRINT#1,A$
80 PRESET(X*16+2,Y*16),,TOR:PRINT#1,A$
90 PRESET(X*16,Y*16+1),,TOR:PRINT#1,A$
100 PRESET(X*16+2,Y*16+1),,TOR:PRINT#1,A$
110 PRESET(X*16,Y*16+2),,TOR:PRINT#1,A$
120 PRESET(X*16+1,Y*16+2),,TOR:PRINT#1,A$
130 PRESET(X*16+2,Y*16+2),,TOR:PRINT#1,A$
140 PRESET(X*16+1,Y*16+1),,XOR:COLOR C2:PRINT#1 A$
150 GOTO 40
160 FOR D=0 TO 2000:NEXT D
170 FOR Y=8 TO 212 STEP 8
180 COPY(0,8)-(255,212),0 TO(0,0),0
190 NEXT Y
200 SET VIDEO 0,0,0,0,0,1,0:END
210 ' DATA
220 DATA 4,1,128,28,"수 퍼 임 포 즈"
230 DATA 11,1,255,120," 란"
240 DATA 2,3,28,255,"비디오나 텔레비젼의 영상"
250 DATA 14,3,255,120,"과"
260 DATA 2,5,35,119,"컴퓨터의 영상"
270 DATA 9,5,255,120,"을"
280 DATA 11,5,55,20,"합성"
290 DATA 13,5,255,120,"하는"
300 DATA 2,7,145,132,"최첨단"
310 DATA 5,7,255,120,"의 기술 입니다"
320 DATA 2,9,255,120,"이 기능으로 비디오 편집등을"
330 DATA 2,11,255,120,"할 수 있읍니다"
340 DATA 0,0,0,0,"##"
```

18. COPY SCREEN

외부 비디오신호를 디지타이즈하여 1/4로 축소한 그림을 만드는 프로그램입니다.
축소한 그림은 BSAVE 혹은 COPY 명령으로 디스켓에 저장할 수도 있읍니다.

```
10 SCREEN 8
20 SET PAGE 0,0:COLOR 255,0,0:CLS
30 SET PAGE 0,1:COLOR 255,0,0:CLS
40 SET VIDEO 3,0,0,1,3,1
50 VDP(7)=255
60 X1=0:Y1=0
70 FOR I=0 TO 7
80 SET PAGE 1,1:SET VIDEO 3,,,1
90 FOR D=0 TO 500:NEXT D
100 COPY SCREEN 0
110 FOR D=0 TO 500:NEXT D
120 SET PAGE 0,1:SET VIDEO 1
130 '
140 FOR X=0 TO 127
150 COPY (X*2+1,0)-(X*2+1,211),1 TO (X,0),1
160 NEXT X
170 FOR Y=0 TO 105
180 COPY (0,Y*2+1)-(127,Y*2+1),1 TO (X1,Y+Y1),0
190 NEXT Y
200 FOR D=0TO 200:NEXT D
210 IF (X1=0) AND (Y1=0) THEN X1=128:Y1=0:GOTO 250
220 IF (X1=128) AND (Y1=0) THEN X1=0:Y1=106:GOTO 250
230 IF (X1=0) AND (Y1=106) THEN X1=128:Y1=106:GOTO 250
240 IF (X1=128) AND (Y1=106) THEN X1=0:Y1=0
250 NEXT I
260 END
```

6

수퍼그래픽

이 장에서는 시스템 디스켓안에 들어있는 그래픽 소프트웨어의 사용방법에 대해서 설명합니다.

1. 시스템의 구성

1. 개요

수퍼 그래픽은 SD (수퍼임포즈, 디지타이즈 보드) 가 내장된 CPC-400S 기종에서 마우스나 키보드를 사용하여 그림을 그리거나 혹은 VTR, 비디오, 카메라 등과 연결하여 여러가지 기능들을 수행할 수 있는 본격적인 그래픽 소프트웨어입니다.

- 외부의 비디오 입력신호를 디지타이즈하여 그 화면을 수정하기도 하고, 그림이나 문자를 첨가할 수 있습니다.
- 정지된 화면을 삼각형으로 모자이크할 수 있습니다.
- 정지된 화면을 반전하거나, 단색화 (흑백사진처럼) 하는 등의 특수효과도 있습니다.
- 비디오 화면에 자막을 집어넣는 텔로프(Telop) 기능, 2 장의 화면을 나누어서 변환시키는 와이프 (Wipe) 기능도 있습니다.

그외에 여러 기능이 있으며, 그려진 그림을 디스켓에 저장할 수도 있고, 프린터로 하드 카피할 수도 있습니다.

※ 본 소프트웨어는 X-Ⅱ 구입시 시스템 디스켓에 같이 포함되어 있습니다.
SD보드가 내장되어 있지 않은 컴퓨터에서는 외부 비디오 처리 기능이 수행되지 않습니다.
또한 시스템 메모리가 128KB이하가 되는 시스템에서는 동작이 되지 않으니 사용하지 마십시오.

2. 필요한 장비

- **컴퓨터** : 수퍼임포즈 기능, 128K byte 이상의 RAM, 한자 ROM이 내장된 것이어야 합니다.
  관련기종 : X-II
- **모니터** : RGB 모니터, 칼라 모니터를 사용할 수 있지만, 보다 선명한 색을 보고 싶으면 RGB 모니터를 사용하십시오.
  관련기종 : CMC-472AW 등
- **마우스** : 커서키로 조작할 수도 있지만 마우스를 사용하는 것이 더욱 편리합니다.
- **비디오카메라** : 화면 입력에 사용합니다.
- **VTR** : 화면입력이나 비디오 편집에 사용합니다.
  본격적으로 비디오 편집을 하려면 2 대가 필요합니다.
- **프린터** : 하드 카피(Hard Copy)를 하고 싶을 때 필요합니다.
  MSX 사양의 프린터가 필요합니다.
  관련기종 : DPP-1400A 등

3. 설치

- 컴퓨터, 모니터는 각각의 취급설명서를 참조하십시오.
- 프린터는 컴퓨터의 프린터 단자와 연결합니다.
- 마우스를 사용할 경우엔 조이스틱 포트1에 연결합니다.
- VTR이나 비디오 카메라 등의 비디오 출력단자를 컴퓨터의 비디오 입력단자에 연결하고, 음성출력단자는 Left와 Right 단자를 구별하여 컴퓨터의 음성 입력단자에 연결합니다.
- 녹화할 경우 컴퓨터의 비디오 출력단자를 VTR의 비디오 입력에 연결합니다.
  VTR에 입력변환 스위치가 있는 경우는 Line IN (외부입력) 으로 세트합니다.

자세한 사항은 해설편의 「수퍼임포즈, 디지타이즈」 항을 참조하여 주십시오.

4. 동작

1. 컴퓨터와 주변 장치와의 연결을 확인합니다.
2. 시스템 디스켓을 A 드라이브에 넣습니다.
3. 컴퓨터 및 주변장치의 전원을 켭니다.
4. MSX-DOS가 부팅되고 "A＞"라는 프롬프트가 나오면 다음과 같이 입력한뒤 [RETURN]키를 칩니다.

   A＞SUPERGRP

5. 화면이 지워지면서 수퍼 그래픽의 초기화면이 표시됩니다.
6. 수퍼 그래픽이 동작됩니다.

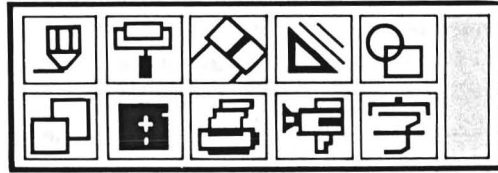
5. 주의사항

수퍼 그래픽의 프로그램이나 하드웨어를 사고로부터 방지하기 위해 다음과 같은 사항을 주의하여 주십시오.

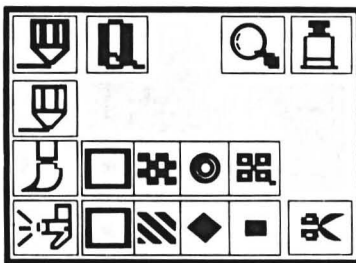
- 수퍼 그래픽이 작동하면 화면에 윈도우와 커서가 표시됩니다. 이 표시가 나타날 때까지는 절대로 시스템 디스켓을 뽑지 마십시오.
- 디스켓에 그림을 저장하거나 읽어내는 도중에 컴퓨터의 전원을 끄지 마십시오.
  또, 이 상태에서 디스켓을 꺼내거나 집어넣지도 마십시오.
- 디스켓 보관시 자기(磁氣) 가 있는 곳에 두지 마십시오.
- 플로피 디스크의 오토셔터를 열어놓거나 디스크면을 직접 만지지 않도록 주의하십시오.

2. 해설

1. 실행방법

수퍼 그래픽을 동작시키면 그림1과 같은 윈도우가 표시됩니다.

(그림1)

윈도우에는 각종의 조작을 나타내는 마크가 있읍니다. 이 마크를 아이콘이라고 합니다.
이 아이콘을 지정하면 그에 해당하는 프로그램이 수행됩니다.
예를 들어 아이콘을 지정하여 실행시키는 방법은 다음과 같읍니다.

1. 마우스(또는 커서키)를 동작시켜 마크를 아이콘 위치에 이동시킨다.
2. 트리거 버튼1(마우스의 왼쪽버튼)이나 스페이스키를 누른다.
3. 그림2와 같은 서브윈도우가 나타난다.
4. 다시 마크를 서브윈도우의 아이콘 위치에 이동시켜 트리거버튼1을 누른다.
5. 그림3과 같은 실행중의 윈도우가 표시된다.(이때에 커서나 마우스로 그림을 그릴 수 있다.)

(그림2)

(그림 3)

참고

- 마우스의 왼쪽 버튼을「트리거 버튼 1」, 오른쪽 버튼을「트리거 버튼 2」라고 합니다.
- 키보드로 그림을 그릴 경우 커서키로 마크를 이동시키고, 트리거 버튼 1대신 스페이스키를, 트리거 버튼 2대신 ESC키를 사용합니다.
- 커서 : 마크를「커서」라고 합니다.
- 커서가 화면상에 표시되어 있을 경우에만 마우스를 사용할 수 있읍니다.
- 클릭 : 트리거 버튼 1을 순간적으로 눌렀다 놓는 것을「클릭」이라고 하고, 트리거 버튼 2를 순간적으로 눌렀다 놓는 것을「2클릭」이라고 합니다.
- 아이콘 : (Quit) 아이콘을 클릭하면 현재 아이콘의 실행을 끝내고 서브윈도우를 표시하거나, 메인 윈도우를 표시합니다.
- 메뉴소거 : 그림 1에서의 맨 우측의 직사각형은 메뉴소거 아이콘으로 이 아이콘을 클릭하면 윈도우가 지워집니다.
  다시 한번 버튼을 누르면 원래 상태로 돌아옵니다.
- 윈도우이동 : 그림을 그릴 때 실행중의 윈도우가 방해될 경우 실행중인 아이콘의 위치에 커서를 놓고 트리거 버튼 1을 누른 채 마우스를 움직이면 윈도우가 따라서 이동됩니다.
- 바로 전에 실행한 것을 지우고 싶을 때 :
  선이나 색칠 등을 잘못하였을 경우 2클릭을 하면 실행전의 상태로 돌아갑니다.
  만약 ①의 그림을 실행한 후 ②의 그림을 실행했을 경우, ②의 그림은 2클릭을 하여 지울 수 있지만 ①의 그림은 이 조작으로는 지워지지 않읍니다. (2클릭을 할 때마다 실행전 화면과 실행후 화면이 번갈아 표시됩니다.)

2. 기능해설

가는 선을 긋는다.

지정된 패턴으로 선을 그린다.

지정된 패턴에 따라 점을 흩트러 뜨리면서 선을 그린다.

지정한 색으로 같은 색의 영역을 칠한다.

막혀진 영역을 칠한다.

지정하는 부분을 지운다.

화면 전체를 지운다.

지정한 두점을 연결하여 선을 긋는다.

원을 그린다.

사각형을 그린다.

지정한 범위의 그림을 그대로 복사한다.

지정한 범위의 그림을 확대, 축소, 90° 단위로 회전 복사한다.

폐곡선으로 지정한 범위의 그림을 복사한다.

화면의 그림을 디스켓에 저장한다.

디스켓에 저장된 그림을 화면에 표시한다.

그림을 흑백으로 하드카피한다.

Mosaic 화면의 그림을 모자이크로 표현한다.

Effect Monotone 지정한 범위의 그림을 흑백으로 표시한다.

Smooth 배경색을 부드럽게 변화시킨다.

Reverse 지정한 범위의 색을 반전한다.

Outline 윤곽선을 표시한다.

Telop 문자열을 화면상의 오른쪽에서 왼쪽으로 이동시키며 표시한다.

Wipe 비디오 입력화면과 컴퓨터 화면을 막을 형성하듯 조금씩 지우면서 교체한다.

Converter 지정한 범위의 지정된 색을 다른색으로 바꾼다.

Generator 화면의 색조를 변화시킨다.

Digitize 비디오 화면을 정지된 화면으로 읽어들인다.

Super on 외부 입력화면과 컴퓨터 화면을 겹쳐서 표시한다.

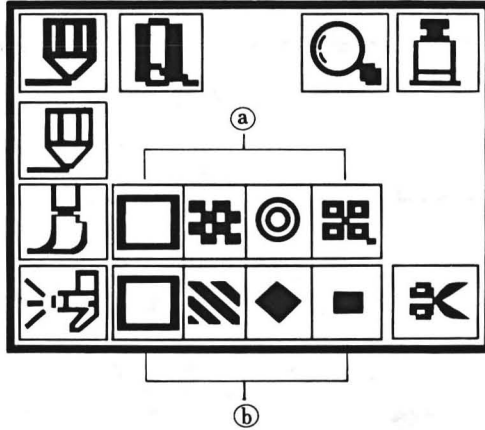
Super off 컴퓨터 화면만을 표시한다.

Code를 입력하여 한자나 특수기호 등을 표시한다.

키보드의 문자를 표시한다.

3. 사용방법

1. 연필

아이콘을 클릭하면 다음과 같은 서브윈도우가 표시됩니다.

● (연필)은 가는 선을 긋는 모드입니다.

실행방법 : 1) 원하는 위치에 커서를 놓고 트리거버튼 1을 누른다.

2) 버튼을 누른채 마우스를 움직이면 거기에 대응하여 에 표시된 색으로 가는선이 그려진다.

3) 트리거버튼 1을 놓고 마우스를 움직이면 커서만 움직인다.

● (붓)은 굵은선을 그리는 모드입니다.

실행방법 : 1) 먼저 패턴ⓐ와 패턴ⓑ에서 원하는 모양을 선택한 후 을 클릭한다.
(패턴의 모양은 ⓐ와ⓑ의 논리연산 AND에 의해 정해진다.)

2) 을 실행하듯 트리거버튼 1을 누른채 마우스를 움직이면 정해놓은 패턴과 에 표시된 색으로 그림이 그려진다.

● (스프레이)는 분무기로 물을 뿌리듯이 정해진 패턴을 흩트러 뜨리면서 그림을 그리는 모드입니다.

실행방법 : (붓)과 같다.

● (돋보기)는 확대모드입니다.

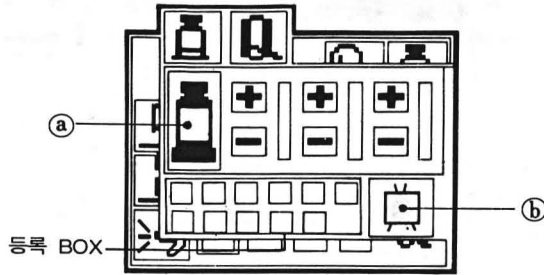
실행방법 : 1) 수정하고 싶은 곳에서 클릭한다.
(클릭된 점을 좌상 꼭지점으로 하여 16×16 화소의 그림이 확대되어 표시된다.)

2) 확대 Box 안에서 원하는대로 클릭하면 그것에 대응한 실제의 위치에도 그림이 표시된다.

3) 확대 Box의 우측에 있는 칼라중 원하는 색을 클릭한후, 확대Box안에서 다시 클릭하면 그 색으로 그림이 그려진다.

4) 확대 Box의 바깥쪽을 클릭하면 그 부분을 확대Box에 표시하므로 연속적으로 수정하는 것이 가능하다.

● (빠레트) 는 색을 지정합니다. 현재 표시된 색은 연필, 페인트, 도형, 선 등을 그릴때 표시되는 색입니다.

아이콘을 클릭하면 다음과 같은 윈도우가 표시됩니다.

○**색지정방법** : 색지정방법은 두가지가 있읍니다.
하나는 빨, 녹, 파 옆에 있는 +, — 기호를 클릭하여 원하는 색의 조합으로 색을 만드는 방법, 또 다른 하나는 이미 화면에 표시되어 있는 색이나 등록Box에 등록되어 있는 색을 꺼내는 방법입니다.

1) **첫번째 지정방법**

① 커서를 빨, 녹, 파 각각의 옆에 있는 +, — 기호에 놓고 클릭하면 색의 막대그래프가 증가 또는 감소된다.

② ⓐ에 표시되는 색이 현재 만들어진 색이므로 이것을 참조하여 원하는 색이 될때까지 ①을 되풀이한다.

2) **두번째 지정방법**

① 커서를 등록Box나 화면에서 원하는 색에 놓고 트리거버튼 1을 누른다.

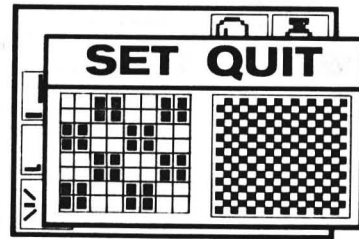
② 버튼을 누른채 ⓐ로 커서를 옮겨놓고 버튼을 놓는다. (원하는 색이 ⓐ에 표시된다.)

○**색등록방법** : 색지정의 두번째 방법과 같읍니다.
예를들어 ⓐ에 표시된 색을 등록시키려면, ⓐ에 커서를 놓고 트리거버튼 1을 누른채 등록시킬 Box의 위치에 커서를 옮긴후 버튼을 놓는다.

○ ⓑ(TV)는 화면의 색을 표시하고 있다. 화면의 색을 바꾸려면 색지정의 두번째 방법을 사용한다. (화면소거를 실행하지 않으면 실제의 화면색은 변하지 않는다.)

● [가위] (가위)는 패턴을 수정하는 것입니다.

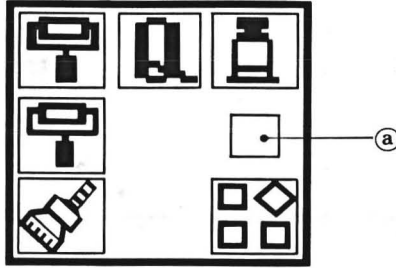
실행방법 : 1) 먼저 패턴ⓐ나 패턴ⓑ에서 변경하고 싶은 패턴을 클릭한다.

2) [가위]을 클릭하면 다음과 같은 윈도우가 나타난다.

3) 8×8의 바둑판 무늬로 패턴이 확대되어 나타난 Box에 커서를 놓고 클릭하면 클릭된 Bit의 내용이 반전된다. (오른쪽 Box의 내용도 대응하여 변하므로 참조하여 주십시오)

4) 원하는 모양으로 패턴이 완성되었으면 [SET]에 커서를 놓고 클릭하고, 패턴변경을 중단할 경우는 [QUIT]에 커서를 놓고 클릭한다.
(전에 표시되었던 서브윈도우로 돌아간다.)

> **참고** 8×8로 나누어져 표시된 칸들을 Bit라고 한다.

2. 페인트

아이콘을 클릭하면 다음과 같은 서브 윈도우가 표시됩니다.

● (롤러붓)은 [빠레트]에 표시된 색과 같은 색으로 닫혀져 있는 부분을 칠합니다. 즉 다른 색으로 막혀져 있는 폐곡선 등의 도형을 무시하고 칠합니다.

실행방법 : 1) 먼저 [에디터]를 클릭하여 원하는 패턴을 선택한다.
2) 선택된 패턴은 ⓐ에 표시된다.
3) [롤러붓]을 클릭한 후 [빠레트]에 표시된 색과 같은 색으로 닫혀져 있는 폐곡선 안에 커서를 놓고 클릭한다. (같은 색으로 되어있는 폐곡선 안을 지정한 패턴으로 메꾼다.)

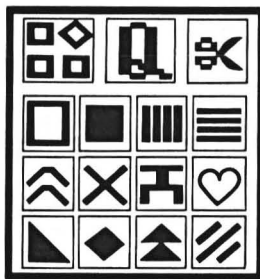
● (솔)은 어느 색으로든 닫혀져 있는 부분을 칠합니다.

실행방법 : 1) 먼저 [에디터]를 클릭하여 원하는 패턴을 선택한 후 [솔]을 클릭한다.
2) 칠하고 싶은 곳에 커서를 놓고 클릭하면 클릭된 점의 색과 다른색을 만나기 전까지의 범위를 [빠레트]에 표시된 색과 ⓐ에 표시된 패턴으로 칠한다.

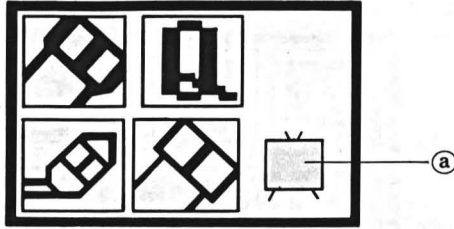
● (빠레트)에는 색을 칠할때의 색을 표시하고 있습니다. 색의 지정방법은 [ ]의 [빠레트]을 참조하십시오.

● ⓐ에는 색을 칠할때의 패턴을 표시하고 있습니다. 패턴의 지정은 [에디터]를 클릭하여 선택합니다.

● (에디터)는 패턴의 모양을 선택하거나 수정하는 모드입니다.

실행방법 : 1) [에디터]를 클릭하면 다음과 같은 윈도우가 나타난다.

2) 원하는 패턴 아이콘에 커서를 놓고 클릭한다. (선택한 패턴의 모양이 반전된다.)
3) 원하는 패턴모양이 없을 경우 [가위]를 클릭하여 패턴수정을 할 수 있다.
(패턴수정방법은 [ ]의 [가위]을 참조하십시오.)
4) [Q]를 클릭하면 전에 표시되었던 서브 윈도우가 표시된다.
(ⓐ의 위치에 선택된 패턴이 표시된다.)

3. 화면소거

아이콘을 클릭하면 다음과 같은 서브 윈도우가 표시됩니다.

● (지우개)는 부분소거 모드입니다.

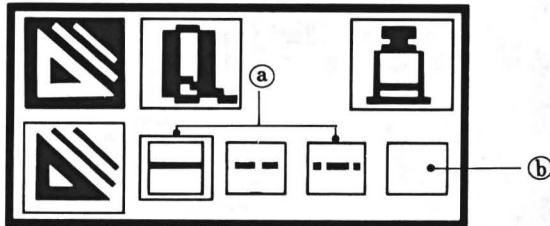
실행방법 : 1) 지우고 싶은 위치에 커서를 놓고 트리거버튼 1을 누른다.
2) 연필로 선을 그리듯이 버튼을 누른채 마우스를 이동시킨다.
(커서의 이동에 대응하여 ⓐ에 표시된 색으로 지워간다.)
3) 부분을 지우는 것이 끝났으면 버튼을 놓는다.

● (화면소거)는 화면전체를 지우는 모드입니다.

실행방법 : 을 클릭한후 또한번 클릭하면 ⓐ에 표시된 색으로 화면을 지운다.

● ⓐ는 화면의 배경색을 표시하고 있읍니다.
ⓐ를 클릭하여 색을 바꿀 수도 있읍니다. 색지정은 의 를 참조하십시오.

4. 직선

아이콘을 클릭하면 다음과 같은 서브 윈도우가 표시됩니다.

● (직선)은 두점을 연결하여 선을 긋는 모드입니다.

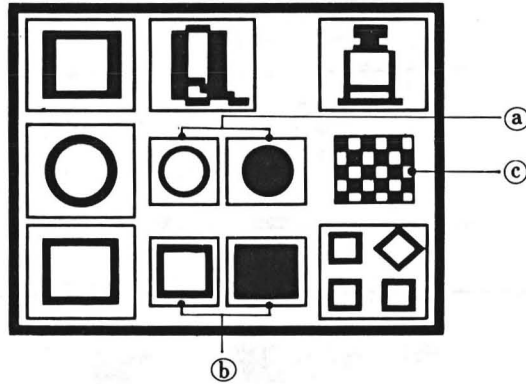
실행방법 : 1) 먼저 ⓐ의 선종류에서 원하는 선을 클릭한다.
2) ⓑ를 클릭하여 선의 굵기를 지정한후 을 클릭한다.
3) 원하는 위치에 커서를 놓고 트리거버튼 1을 누른다. (시작점지정)
4) 버튼을 누른채 마우스를 이동하여 원하는 위치에서 버튼을 놓는다. (끝점 지정)
시작점과 끝점을 에 표시된 색과 ⓐ에서 선택한 선으로 이어준다.

● (빠레트)에는 직선을 그을때 나타내는 색을 표시하고 있읍니다. 색의 지정방법은 의 를 참조하십시오.

● ⓐ는 선의 종류를 표시하고 있읍니다.
왼쪽부터 직선, 점선, 이점쇄선의 순으로 표시되어 있읍니다.
원하는 선의 종류를 클릭하면 색이 반전되어 표시됩니다.

● ⓑ는 선의 굵기를 지정하는 것입니다.
ⓑ를 클릭할 때마다 ⓐ에서 선택되어 있는 선이 1라인씩 두꺼워집니다. 굵기는 최대가 8라인으로써, 선의 굵기가 최대로 되어있을때 ⓑ를 클릭하면 초기상태(1라인)로 됩니다.

5. 도형

아이콘을 클릭하면 다음과 같은 서브 윈도우가 표시됩니다.

● (원)은 원을 그리는 모드입니다.

실행방법 : 1) 먼저 ⓐ에서 속이 칠해지지 않은 원을 그릴 것인지, 칠해진 원을 그릴 것인지를 선택한다. (ⓐ의 왼쪽 아이콘은 빈원을 그리는 것이고, 오른쪽 아이콘은 칠해지는 원을 그리는 것이다.)

2) ⓐ에서 오른쪽 아이콘을 선택하였을 경우 를 클릭하여 원하는 패턴을 선택한다.
3) 을 클릭하면 원을 그리는 모드로 된다.
4) 먼저 원하는 위치에 커서를 놓고 트리거버튼 1을 누른다. (중심점 지정)
5) 버튼을 누른 채로 마우스를 이동하여 원하는 원의 크기가 되면 버튼을 놓는다. (반지름 길이 지정)
에 표시된 색으로 원이 그려진다. (ⓐ에서 오른쪽 아이콘을 선택하였을 경우 ⓒ에 표시된 패턴으로 원이 메꾸어진다.)

● (사각형)은 사각형을 그리는 모드입니다.

실행방법 : 원을 그리는 방법과 비슷하다.

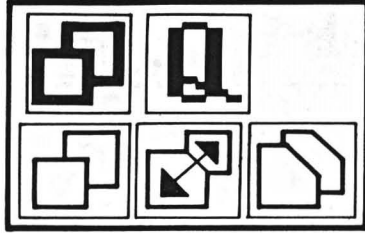
1) 먼저 ⓑ에서 빈사각형을 그릴 것인지, 메꾸워진 사각형을 그릴 것인지 선택한다.
2) 메꾸워진 사각형을 그릴 경우 를 클릭하여 패턴을 선택한다.
3) 사각형을 그리는 방법은 직선을 그리는 방법과 같다. 즉 원하는 위치에서 트리거버튼 1을 누른채로 마우스를 움직여 원하는 사각형의 크기가 되었을때 버튼을 놓는다.
4) 에 표시된 색으로 사각형이 그려진다. (ⓑ에서 오른쪽 아이콘을 선택했을 경우 ⓒ에 표시된 패턴으로 사각형이 메꾸어진다.)

주의 메꿔주는 원이나 사각형을 그릴때, 그 영역안에 그려지는 같은 색이 있을 경우 ⓒ에 표시된 패턴으로 원전체를 메꾸지 못하는 경우가 있읍니다.

● (빠레트)에 표시된 색은 원이나 사각형을 그릴때 나타내는 색입니다. 색의 지정방법은 의 을 참조하십시오.

● ⓒ에 표시된 패턴은 메꾸는 원이나, 사각형을 그릴때 칠해지는 패턴입니다.
를 클릭하여 ⓒ에 표시된 패턴의 모양을 바꿀 수 있읍니다. 패턴의 모양을 바꾸거나 수정하는 방법에 대해서는 의 를 참조하십시오.

6. 복사

아이콘을 클릭하면 다음과 같은 서브 윈도우가 표시됩니다.

● (1:1복사)는 원래의 모습 그대로 복사합니다.

실행방법 : 1) 사각형을 그리는 요령으로 복사할 영역을 지정한다.
2) 커서를 복사시킬 위치에 놓고 클릭한다. (사각형으로 지정한 영역의 그림이 그대로 복사된다.)

> **주의** 사각형으로 지정한 영역의 크기가 복사시키기 위해 클릭한 곳부터 오른쪽으로 남은 화면의 범위보다 크면 클릭한 위치에서 왼쪽으로 옮겨져 복사됩니다.

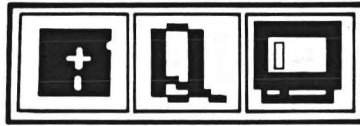
● (1 : n복사)는 확대, 축소, 90° 단위로 회전 복사합니다.
축소나 확대 복사를 할때 가로와 세로의 비율의 차가 커도 괜찮습니다.

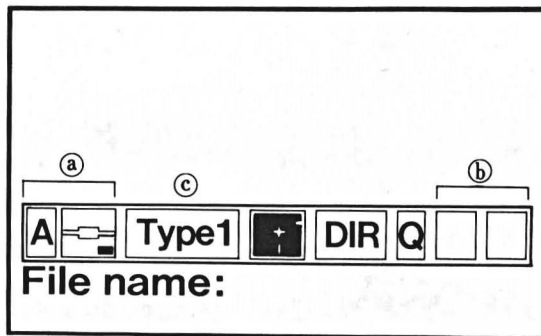
실행방법 : 1) 사각형을 그리는 요령으로 복사할 영역을 지정한다.
2) 1. 축소 복사를 하려면 복사시킬 영역을 지정할때, 복사할 영역의 크기보다 작게 지정한다.
2. 확대 복사를 하려면 복사시킬 영역을 지정할때, 복사할 영역의 크기보다 크게 지정한다.
3. 회전시키지 않고 복사하려면 복사시킬 영역을 지정할때, 왼쪽 위를 시작점으로 하고, 오른쪽 아래를 끝점으로 하여 사각형을 그린다.
4. 90° 회전(오른쪽으로 회전)시켜 복사하려면 복사시킬 영역을 지정할때, 오른쪽 위를 시작점으로 하고, 왼쪽 아래를 끝점으로 하여 사각형을 그린다.
5. 180° 회전(위·아래 역전)시켜 복사하려면 복사시킬 영역을 지정할때, 오른쪽 아래를 시작점으로 하고 왼쪽 위를 끝점으로 하여 사각형을 그린다.
6. 270° 회전(왼쪽으로 회전)시켜 복사하려면 복사시킬 영역을 지정할때, 왼쪽 아래를 시작점으로 하고 오른쪽 위를 끝점으로 하여 사각형을 그린다.

● (부정형 복사)는 복사할 영역을 사각형으로 지정하지 않고 자유로운 폐곡선으로 지정하는 것입니다.

실행방법 : 1) 먼저 의 서브 윈도우에서 에 표시된 색을 확인한 후 복사할 영역중에 없는 색을 지정한다. (복사할 영역중에 있는 색으로 지정하면 복사가 제대로 되지 않는다.)
2) 을 클릭한 후 복사할 영역의 둘레를 지정한다. (둘레는 연필로 선을 그리는 요령으로 폐곡선을 그린다.)
3) 복사시킬 위치에 커서를 놓고 클릭하면 원래의 모양이 그대로 복사된다.

7. 디스켓

아이콘을 클릭하면 다음과 같은 서브 윈도우가 표시됩니다.

- (세이브)나 (로드)를 클릭하기 전에 디스크 드라이브에 그림 데이타 보관용 디스켓이 들어있는지 확인하십시오.

- (세이브)는 화면에 표시되어 있는 그림을 디스켓에 보존하는 모드입니다.
  를 클릭하면 다음과 같은 윈도우가 나타납니다.
  (디스켓에 그림 데이타가 보존되어 있는 경우 아이콘의 위쪽에 화일명들이 표시됩니다.)

○ⓐ는 사용할 디스크 드라이브명을 표시하고 있읍니다.
실행방법 : 1) ⓐ를 클릭하면 사용할 디스크 드라이브가 B임을 표시해준다.
2) 다시 한번 클릭하면 A를 표시해준다.

○ⓒ는 디스켓에 그림을 저장할때 임의의 부분을 저장할 것인지, 아니면 화면의 전체를 저장할 것인지를 선택하는 것입니다. 즉, Type 1일 경우는 화면전체를, Type 2일 경우는 정해진 영역을 디스켓에 저장할 수 있읍니다.
실행방법 : ⓐ를 클릭하여 드라이브명을 교체하듯이 ⓒ를 클릭하여 Type을 바꾼다.
(클릭할 때마다 Type 1과 Type 2를 번갈아 표시한다.)

○ 은 「File name :」 옆에 표시되어 있는 화일명으로 그림을 저장하게 하는 것입니다.
1) 저장할 화일명을 지정하지 않았을 경우 을 클릭하여도 그림이 저장되지 않는다.
2) 지정한 화일명이 이미 디스켓에 있을때 을 클릭하면 「Overlay Ok ? YES NO」라는 메시지가 나타난다.
1. 먼저 있던 데이타 화일을 무시하고 새로운 그림을 저장할 때는 「Y」 키를 누르던지, YES에 커서를 놓고 클릭한다.
2. 먼저있던 데이타 화일을 지우지 않으려면 「N」 키를 누르던지 NO에 커서를 놓고 클릭한다.
3) Type 1일 경우 「GR1」이, Type 2일 경우 「GR2」가 지정한 화일명 뒤에 타잎명으로 붙여져서 디스켓에 보존된다.

실행방법 : 1) Type 1로 세트되어 있을 경우 [■]를 클릭하면 화면전체가 디스켓에 보존된다.
2) Type 2로 세트되어 있을 경우 [■]를 클릭한 후 사각형을 그리는 요령으로 디스켓에 저장할 영역을 지정한다. 클릭을 하면 디스켓에 저장을 하고, 2 클릭을 하면 다시 영역을 지정하게 한다.
3) 디스켓에 그림이 다 기록되었으면 화면에 메인 윈도우가 나타난다.

주의
- 플로피 디스크의 메모리가 부족하면 「Disk full」이라는 에러 메시지가 표시됩니다. 이럴땐 디스켓을 바꿔넣고 다시 [■] 를 클릭하십시오.
- 화일명이 잘못되어 있을 경우 「Bad file name」이라는 에러메시지가 표시됩니다. 이럴땐 화일명을 다시 입력한 후 세이브 명령을 실행시키십시오.

○ [DIR]은 디스켓에 보존되어 있는 그림 데이타 화일명을 표시해줍니다.
실행방법 : [DIR] 아이콘을 클릭하면 ⓐ가 표시하고 있는 디스켓의 화일명들이 표시된다. 이경우 ⓒ에 지정된 타잎의 화일명이 표시된다. 화일이 6개 이상이면 ⓑ에 「↑」가 표시된다.

○ [Q]를 클릭하면 메인 윈도우로 돌아갑니다.

○ⓑ에 「↑」나 「↓」 표시가 되어있을 경우, 그것을 클릭하면 표시되어 있는 화일명이 상하로 스크롤 됩니다.

● [로드 아이콘] (로드)는 디스켓에 보존되어 있는 그림을 화면에 표시하는 모드입니다.
[로드 아이콘] 를 클릭하면 다음과 같은 윈도우가 나타납니다.

(즉, [로드 아이콘] 를 제외하면 모든 것이 세이브 윈도우와 같습니다.)

○ [로드 아이콘] 는 「File name : 」 옆에 표시되어 있는 화일명으로 디스켓에 저장되어 있는 그림을 화면에 표시하게 하는 것입니다.

6 수퍼그래픽

○읽어낼 화일명이 지정되어 있지 않을 때는 [icon]를 클릭하여도 디스켓에서 그림을 읽어내지 않습니다.

실행방법 : 1) [icon] 를 클릭하면 「Logical ? Tpset Pset」이라는 메시지가 표시된다.
2) 색상의 논리연산(지정색이 투명색, 즉 0이면 현재색을, 아니면 지정색을 표시)을 하여 로드하고 싶으면 [Tpset]에 커서를 놓고 클릭하던지, 「T」 키를 누른다. 그렇지 않을 경우 [Pset]에 커서를 놓고 클릭하던지 「P」 키를 누른다.
3) Type 1로 지정되어 있을 경우 2)의 실행만으로 디스켓에 들어있는 그림이 화면에 표시된다.
4) Type 2로 지정되어 있을 경우 로드하고 싶은 위치에 커서를 놓고 클릭하면 그 위치에 그림이 표시된다. 원하는 위치에 로드되지 않았으면 2클릭을 하여 로드 위치를 다시 지정한다. 원하는 위치에 로드되었으면 클릭을 한다.
5) 로드가 끝나면 메인 윈도우로 돌아간다.

[주의] ● 데이타가 잘못 기록되어 있을 경우 「Read error ⁄」라는 에러 메시지가 나타납니다.
● 지정한 화일명이 디스켓에 저장되어 있지 않으면 「File not found」라는 에러 메시지가 나타납니다.

○화일명은 8자이내의 영숫자이어야 합니다. (단, 한글은 화일명으로 사용할 수 없읍니다.)

○리턴키를 치면 실행 아이콘([icon] 과 [icon])을 클릭한 것과 같은 효과가 나타납니다.

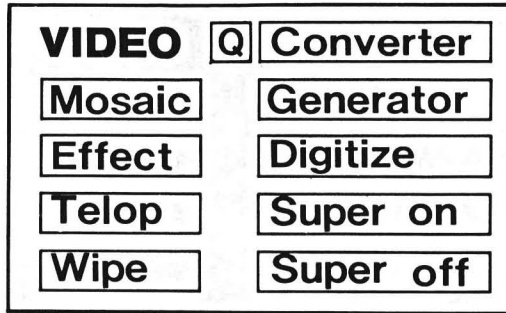
○이미 디스켓에 저장되어 있는 화일명을 지정할 때는 키보드로 입력하여도 관계없읍니다만, 커서를 윈도우에 표시된 화일명에 놓고 클릭하여도 됩니다.
(「File name : 」 옆에 화일명이 표시됩니다.)

8. [icon] 프린터

[icon] 는 화면에 표시되어 있는 그림을 인쇄하는 것입니다.
이 아이콘을 클릭하면 인쇄를 시작합니다. 인쇄가 끝나면 메인윈도우로 돌아갑니다.

[주의] ● 프린터가 연결되어 있지 않은 상태에서 이 아이콘을 클릭하였을 경우 약간의 시간이 경과되면 저절로 메인윈도우로 돌아간다.
● 프린터 상태가 좋지않아 중지하고 싶으면 [CTRL]+[STOP] 키를 누르십시오.

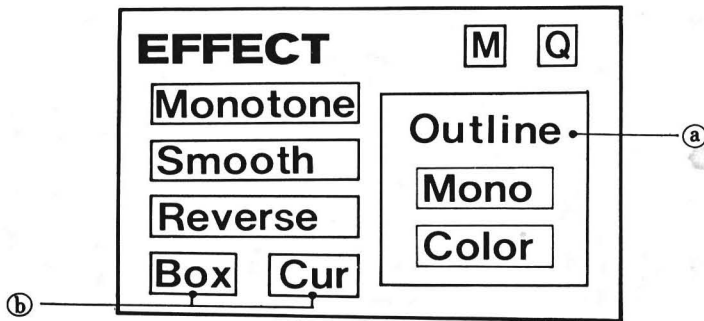
9. 비디오

아이콘을 클릭하면 아래와 같은 서브 윈도우가 표시됩니다.

● Mosaic (모자이크)는 현재의 화면에 표시되어 있는 그림을 「△」의 패턴으로 모자이크합니다.

실행방법 : 1) Mosaic 를 클릭하면 서브 윈도우가 지워지면서 현재 화면에 있는 그림이 모자이크 된다.

2) 이 상태에서 2클릭하면 이전의 그림이 나타나고, 다시 2클릭하면 모자이크된 그림이 나타난다. (2클릭할 때마다 실행전 화면과, 실행후 화면이 번갈아 표시됩니다.)

3) 트리거버튼 1을 누르면 원래의 서브 윈도우로 되돌아간다.

● Effect (이펙트)는 현재 화면에 표시되어 있는 그림에 특수한 효과를 처리해주는 기능입니다. Effect 를 클릭하면, 다음과 같은 윈도우가 표시됩니다.

실행방법 : 1) Monotone, Smooth, Reverse, ⓐ중 처리할 기능을 한가지 설정한다.

2) ⓑ의 Box, Cur 중 하나를 선택하여 영역을 설정한후, 클릭하면 앞에서 지정한 기능이 실행된다.

3) 실행이 끝난후, 클릭을 하면 Effect 윈도우 화면으로 돌아간다.

설　명 : 1) Monotone 은 지정된 범위의 색을 흑백으로 처리해준다.

2) Smooth 는 지정된 범위의 색을 부드럽게 해준다.

3) Reverse 는 지정된 범위의 색을 반전한다.

4) ⓐ는 지정된 범위안에 있는 도형들의 윤곽선을 표시해주는 것으로, Mono 와 Color 가 있다.

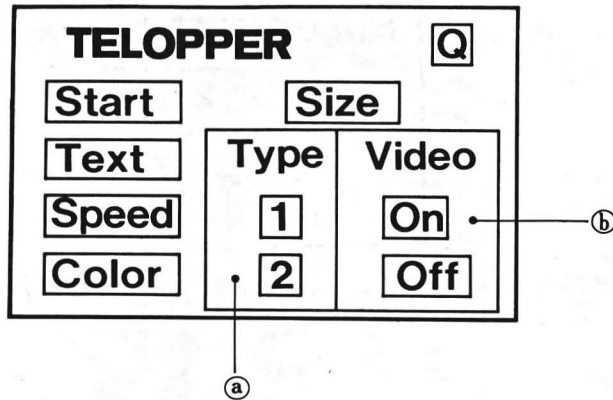
1. Mono 는 윤곽선을 하얀색으로 표시해준다.
2. Color 는 윤곽선을 도형색으로 표시해준다.

주의 도형의 형태에 따라 윤곽선 색이 도형의 색과 다르게 표시될 수도 있읍니다.

5) ⓑ는 실행 아이콘으로, [Box]를 클릭한 경우에는 범위를 사각형으로 지정한 후, 클릭하면 지정된 범위가 앞에서 지정한 아이콘으로 실행된다.
[Cur]를 클릭한 경우에는, 범위를 곡선으로 지정한후 클릭하면, 일단은 지정된 범위를 곡선의 색으로 한번 칠한 후에 앞에서 지정한 아이콘을 실행한다.
(곡선의 색지정은 [ ]의 색지정을 참조하십시오.)

6) [M]을 클릭하면 현재의 윈도우가 사라지고, 다시 클릭하면 윈도우가 표시된다.

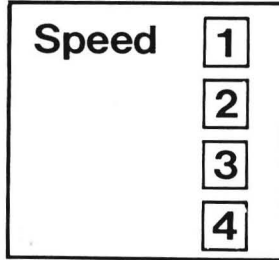
주의 곡선으로 범위를 지정한 후, 클릭할 경우, 커서를 지정된 범위안으로 옮겨놓은 후 눌러야 선택된 아이콘이 실행됩니다.
만약, 지정된 범위 밖 또는 범위의 경계선에 커서를 놓고 클릭하면 지정된 범위는 지워지므로, 다시 범위를 지정해야 합니다.

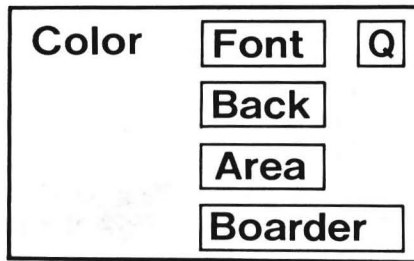
● [Telop] (텔로프)는 화면에 자막처리를 해주는 기능입니다.
[Telop]를 클릭하면 다음과 같은 윈도우가 표시됩니다.

실행방법 : 1) ⓐ, ⓑ, [Text], [Speed], [Color], [Size]를 설정한후 [Start]를 클릭하여 실행시킨다.
2) 실행이 끝난 후 클릭이나 2클릭을 하면 [Telop] 윈도우 화면으로 돌아간다.

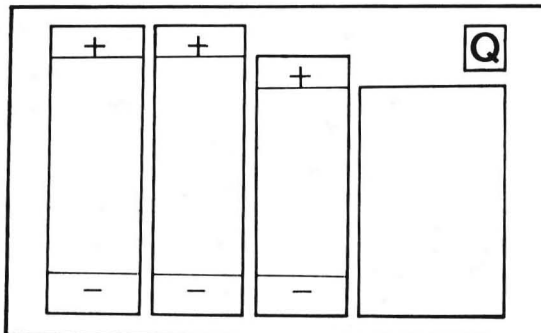
○ [Text]를 클릭하면 텔로프 윈도우가 사라지고, 커서가 화면의 좌측 상단에 표시된다.

1) 화면의 하단에 「Key input」이라는 메시지가 표시된 상태에서 [SELECT] 키를 한번 누르면 「Code:」라는 메시지가 화면에 표시된다.
[SELECT] 키를 누를 때마다 「Key input」과 「Code:」 라는 메시지가 번갈아 화면에 표시된다.
한번 선택된 모드는 다시 선택할 때까지 그 모드상태를 유지한다.
2) 「Key input」 메시지상태 : 키보드상의 문자를 입력할 수 있는 상태이다.
3) 「Code:」 메시지상태 : 코드를 입력하여 한자나 특수기호를 표시할 수 있는 상태이다.
4) [SPACE] 키나 [BS] 키 또는 커서키 [←]를 사용하여 문자를 수정할 수 있다.
[RETURN] 키를 누르면 화면에는 "@"가 표시되는데 이것은 Type [2]에서는 공백으로 처리되고, Type [1] 에서는 이전에 텔로프된 문자열을 위로 스크롤시킨다.
5) [ESC] 키를 누르면 입력을 종료하고 [Telop] 윈도우로 되돌아간다.

주의 텔로프 문자열을 작성할때 현재 커서가 있는 위치까지만 실행되니 주의하십시오.

○ [Speed]는 문자가 흘러가는 속도를 지정해주는 기능입니다.
[Speed]를 클릭하면 아래와 같은 윈도우가 표시됩니다.

원하는 번호를 클릭하면 텔로프 속도가 설정되고 [Telop] 윈도우로 돌아간다. 이때 [4]는 가장 빠르고, [1]이 가장 늦은 속도로 설정된다.

○ [Color]는 글자색, 바탕색, 주변색을 지정해주는 기능입니다.
[Color]를 클릭하면 다음과 같은 윈도우가 표시됩니다.

1) [Font] 는 화면에 표시되는 글자의 색을 지정한다.
2) [Back] 은 바탕색을 지정한다.
3) [Area] 는 영역색을 지정한다.
4) [Boarder] 는 주변색을 지정한다.

○ [Font], [Back], [Area], [Boarder] 중 하나를 클릭하면 다음과 같은 윈도우가 표시된다.

좌측의 빨간색, 녹색, 파란색을 +, −로 적당히 혼합하면 오른쪽에 합성된 색이 표시된다.

○ [Size] 는 문자가 텔로프되는 범위를 지정해주는 기능입니다. 이것을 클릭하면 윈도우는 지워지고, 화면상에 커서가 나타납니다.

1) 사각형을 그리는 요령으로 텔로프할 범위를 지정한 후, 클릭하면 범위가 지정되면서 [Telop] 윈도우 화면으로 돌아간다.
2) 표시되는 문자 크기는 16×16이다.

○ⓐ는 문자를 텔로프할 때, 표시방법을 지정해 주는 기능입니다.

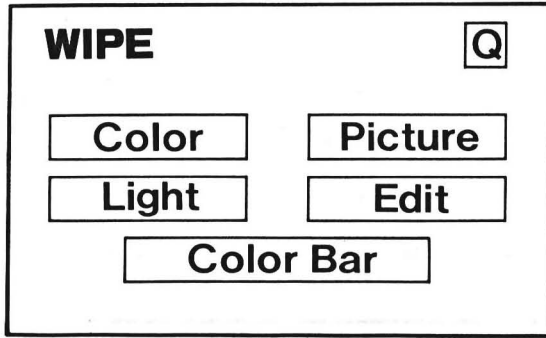
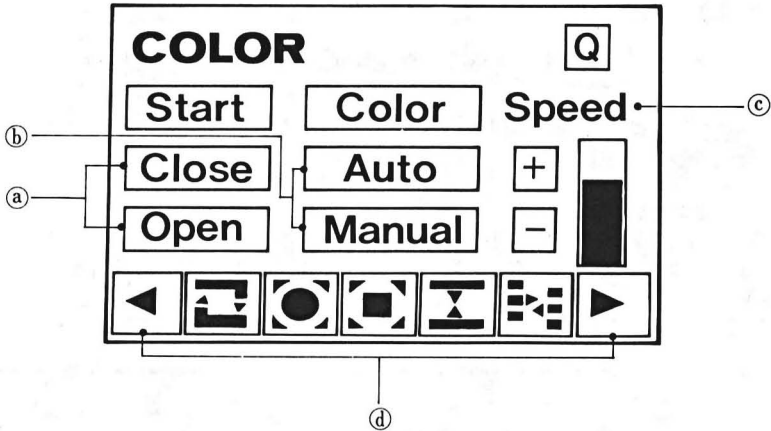
1) Type [1]은 문자열을 오른쪽에서 왼쪽으로 표시하다가 문자열중에 있는 제어문자 [RETURN] 을 만나면, 그 행의 문자열이 위로 올라가고, 만약 다음 행이 있으면, [RETURN] 을 만날 때까지 오른쪽에서 왼쪽으로 문자를 이동해가며 표시해간다.
   마지막 행까지 문자가 표시되면, 이제까지 표시된 문자들이 범위의 최상단으로 올라가면서 표시된다.
2) Type [2]는 문자열을 오른쪽에서 왼쪽으로 이동해가면서 표시한다. 여기서는 문자열 중에 있는 [RETURN] 은 공백으로 처리된다.
3) 텔로프 실행도중 트리거버튼1이나 [SPACE] 키를 누르고 있으면 그 키가 떼어질 때까지 이동중이던 문자열이 멈추고 있게 된다.

○ⓑ(비디오)는 비디오 신호의 입력을 제어해주는 기능입니다.

1) [On]을 클릭하면 외부 비디오 신호가 입력되어 컴퓨터 화면과 합성된 다음 표시된다.
2) [Off] 를 클릭하면 외부 비디오 신호의 입력을 중단시키고 원래의 컴퓨터 화면을 표시한다.
3) [On] 상태에서 [Q]를 클릭하면 외부 비디오 신호의 입력을 중단하고 [아이콘] 윈도우 화면으로 돌아간다.

○ [Start] 를 클릭한 후, 다시 한번 클릭하면 앞에서 지정된 기능들에 의해 텔로프가 실행된다.
실행이 끝난후, 클릭 또는 2 클릭을 하면 [Telop] 윈도우 화면으로 돌아간다.

> **[주 의]** [Speed], [Color], [Size], ⓐ, ⓑ는 다시 선택하지 않는한 그대로 보존된다. 단 [Speed], ⓐ, ⓑ는 [아이콘] 윈도우의 [Q] 를 클릭하면 초기상태로 된다.

● [Wipe]는 외부비디오 입력화면과 컴퓨터 화면을 지정된 패턴으로 교체해 줍니다.
[Wipe]를 클릭하면 다음과 같은 윈도우가 표시됩니다.

○ [Color] 와이프는 비디오 입력화상과 사용자가 선택한 단색의 화면을 바꿔주는 기능입니다.
[Color]를 클릭하면 아래와 같은 윈도우가 표시됩니다.

실행방법 : 1) [Color], ⓐ, ⓑ, ⓒ, ⓓ를 각각 지정하여 패턴의 종류, 속도 등을 지정한다.
2) [Start]를 클릭하면 [Color] 와이프가 실행된다.
3) 실행이 끝난 후, 클릭 또는 2클릭을 하면, [Color] 와이프 윈도우 화면으로 돌아간다.

설 명 : 1) [Color]는 컴퓨터 화면의 색을 지정하는 기능이다.
[Color]를 클릭하면 텔로프의 칼라에서 [Font] 윈도우와 같은 윈도우가 표시된다.
사용방법도 같으므로 참조해 주십시오. 컴퓨터 화면의 색지정은 [Wipe] 윈도우의 [Light]에서도 지정할 수 있다.

2) ⓐ는 어느 화면에서 어느 화면으로 바뀔 것인가를 지정하는 기능이다.
[Close]는 비디오 입력화면에서 컴퓨터 화면으로 바뀐다. (초기상태의 값이다.)
[Open]은 컴퓨터 화면에서 비디오 입력화면으로 바뀐다.

3) ⓑ는 [Color] 와이프 실행을 자동으로 할 것인가, 수동으로 할 것인가를 지정하는 기능이다.
[Auto]는 [Color] 와이프를 자동으로 실행시킨다. (초기상태의 값이다.)
[Manual]은 [Color] 와이프를 자동으로 실행시킬 때 한 블럭마다 클릭을 해줘야 한다.

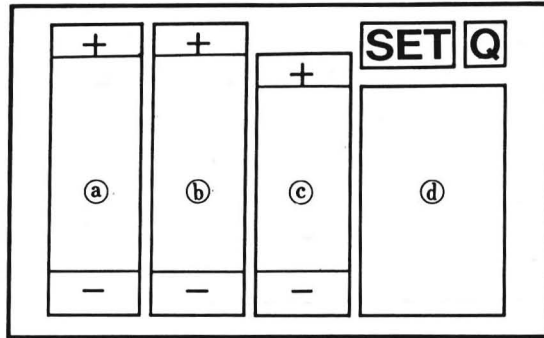
4) ⓒ는 실행속도를 지정하는 기능으로, +를 클릭하면 속도가 빨라지고, -를 클릭하면 속도가 느려진다.

5) ⓓ는 [Color] 와이프를 실행할 때 화면의 형상을 정하는 것으로 9가지의 패턴이 있다.

1. 원하는 패턴에 커서를 놓고 클릭하면 패턴이 반전된다.
2. 화면상에는 5개의 패턴만이 표시되어 있으며, [◀] 또는 [▶]를 클릭하여 9종류의 패턴을 화면상에 표시할 수 있다.
3. 패턴 중에서 [U S] 은 [Wipe] 윈도우에 있는 [Edit]에서 사용자가 작성한 패턴을 사용할 수 있는 패턴이다.

6) [Start]를 클릭한 후, 다시 한번 클릭하면 [Color] 와이프가 실행된다.

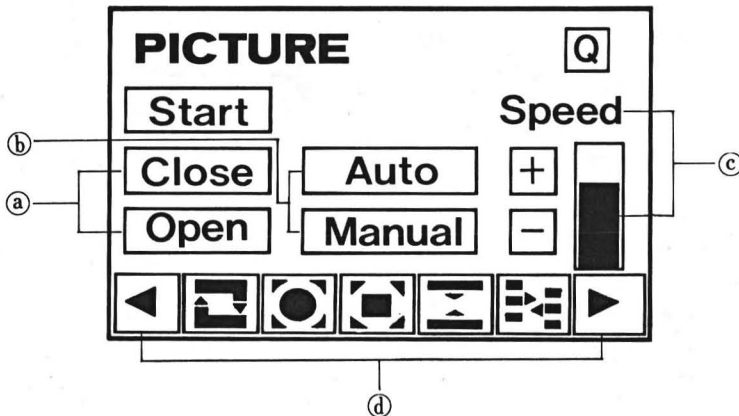
[주의] ● [Color]는 다시 지정하거나, [Light]에 다시 지정하지 않는한 그대로 보존된다.
● ⓐ, ⓑ, ⓒ, ⓓ는 다시 지정하거나, [Picture] 와이프에서 다시 지정하지 않는한 그대로 보존된다. 단, ⓐ, ⓑ는 [아이콘] 윈도우의 [Q]를 클릭하면 초기상태로 된다.

○ Light 는 Color 와이프에서 사용할 색을 설정하는 기능입니다.
Light 를 클릭하면 다음과 같은 윈도우가 표시됩니다.

실행방법 : 실행방법은 텔로프의 칼라에서 Font 윈도우와 같으므로 그것을 참조해 주십시오.

설 명 : 1) ⓐ, ⓑ, ⓒ, ⓓ 각각의 기능은 텔로프의 칼라에서 Font 윈도우와 같으므로 그것을 참조해 주십시오.

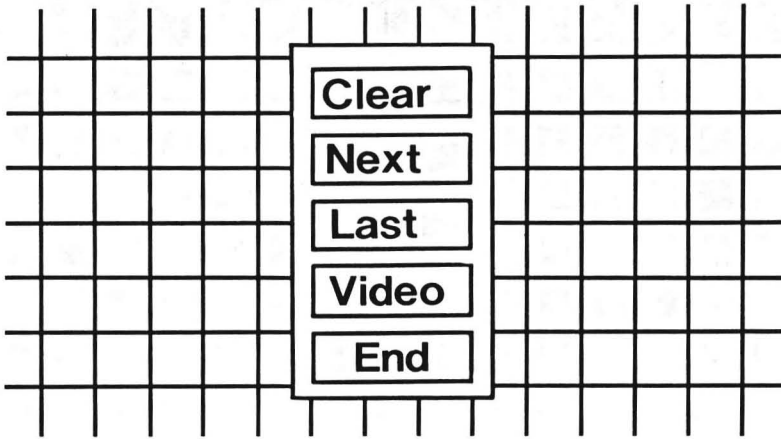
2) SET 를 클릭하면 Light 윈도우가 사라지면서 ⓓ에 표시되어 있는 색이 화면에 표시된다. 다시 한번 클릭하면 Light 윈도우로 돌아온다.

○ Color Bar 는 화면의 색상조정을 위한 기능으로, Color Bar 를 클릭하면, 색상의 테스트 패턴이 표시됩니다.
또 한번 클릭하면 Wipe 윈도우로 돌아갑니다.

○ Picture 와이프는 비디오 입력화상과 사용자가 작성한 그래픽 화면을 바꿔주는 기능입니다.
Picture 를 클릭하면 다음과 같은 윈도우가 표시됩니다.

실행방법 : 실행방법은 앞에서 설명한 Color 와이프와 같으므로 그것을 참조해 주십시오.

○ [Edit]는 [Color] 와이프와 [Picture] 와이프에서 사용하는 화면형상의 패턴을 사용자가 독자적으로 작성할 수 있게금 되어 있는 기능입니다.
[Edit]를 클릭하면 [Wipe] 윈도우가 사라지고, 화면전체가 사각형으로 나뉘어 집니다.
사각형 하나는 1비트를 의미합니다.

실행방법 : 1) 원하는 비트에 커서를 옮겨 놓은 후 클릭하면, 그 비트가 녹색으로 변하고, 다시한번 클릭하면 원래의 상태로 돌아온다.

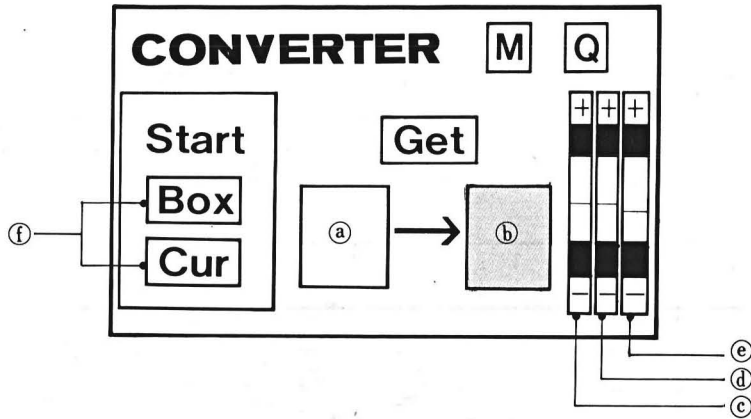
2) 2클릭을 하면 다음과 같은 윈도우가 표시된다.

3) [Clear]를 클릭하면 화면이 지워지고, 초기상태의 [Edit] 화면으로 돌아간다.

4) [Next]를 클릭하면 [Edit] 화면으로 돌아가서, 녹색의 비트들을 파란색으로 바꿔준다. 이것을 계속 반복하여 [Next]를 클릭한 비트들간에 시간차를 준다.

5) [Last]는 [Next] 실행이 끝난 비트들을 수정하고 싶을 때 사용하는 기능이다. 즉, [Last]를 클릭할 때마다 [Edit] 화면으로 돌아가서 [Next]가 실행된 비트들의 역순으로 파란색을 녹색으로 바꿔준다.

6) [Video]를 클릭하면 [Edit] 화면에 비디오 입력화상이 표시된다. 다시 한번 클릭하면 비디오 입력화상이 해제된다.

7) [End]를 클릭하면 [Wipe] 윈도우로 돌아간다.

● [Converter] (컨버터) 는 지정한 범위의 지정된 색을 원하는 색으로 바꿔주는 기능입니다.
[Converter] 를 클릭하면 다음과 같은 윈도우가 표시됩니다.

실행방법 : 1) [Get] 과 ⓒ, ⓓ, ⓔ 를 사용하여 ⓐ, ⓑ 를 설정한다.

2) ⓕ의 [Box], [Cur] 중 하나를 선택하여 영역을 설정한 후, 클릭하면 지정된 범위안에 있는 색중 ⓐ의 색이 모두 ⓑ의 색으로 바뀌어진다.

3) 실행이 끝난 후, 클릭을 하면 [Converter] 윈도우 화면으로 돌아간다.

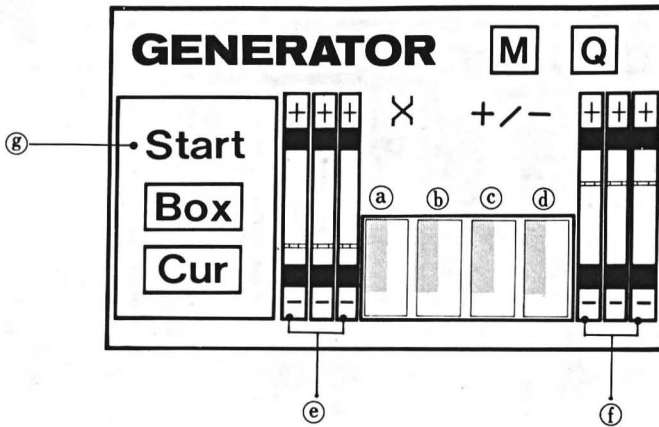
설　명 : 1) [Get] 은 바꾸고 싶은 색을 선택하는 것이다. [Get] 을 클릭하면 윈도우가 사라지고 그래픽 화면이 표시된다. 그래픽 화면에서 원하는 색에 커서를 옮겨 놓은 후 클릭하면, [Converter] 윈도우가 표시되면서 ⓐ에 색이 표시된다.

2) ⓒ, ⓓ, ⓔ는 각각 빨간색, 녹색, 파란색을 표시해준다. ⓒ, ⓓ, ⓔ에 있는 +, −를 각각 클릭하여 원하는 색을 만들면 ⓑ에 그 색이 표시된다.

3) ⓕ는 [Effect] 의 ⓑ와 같으므로 그것을 참조하십시오.

1. ⓕ를 실행하면, 지정된 범위에 있는 ⓐ의 색이 ⓑ의 색으로 바뀌진다.
2. 지정된 범위내에 ⓐ의 색이 없으면, 실행후에도 아무런 변화가 없다.

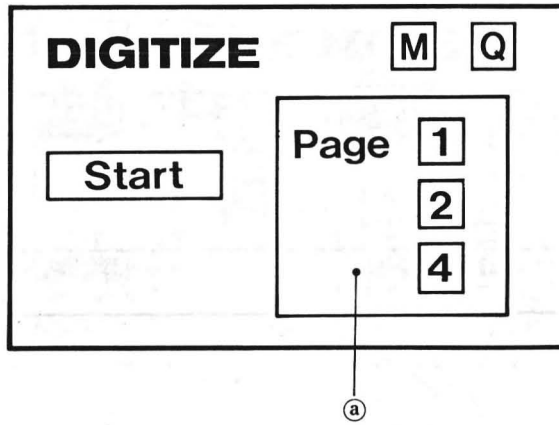
주의 ● ⓐ와 ⓑ의 색을 [Get]이나 ⓒ, ⓓ, ⓔ를 이용하지 않고 선택할 수도 있다.
즉 [Converter] 윈도우가 표시된 상태의 화면에서 바꾸고 싶은 색에 커서를 옮겨 놓은 후, 트리거 버튼 1을 누른 상태로 커서를 이동시켜 ⓐ나 ⓑ에 옮겨 놓은 후, 버튼에서 손을 떼면 선택된 색이 ⓐ나 ⓑ에 표시된다.
단, 윈도우 안에 있는 색은 선택되지 않는다.
● ⓐ와 ⓑ는 다시 지정하지 않는한 그대로 보존된다.

● [Generator] (제너레이터) 는 지정된 범위를 지정된 색으로 바꿔주는 기능입니다.
[Generator] 를 클릭하면 다음과 같은 윈도우가 표시됩니다.

실행방법 : 1) ⓔ와 ⓕ를 이용하여 ⓐ, ⓑ, ⓒ, ⓓ를 설정한다.
2) ⓖ의 [Box] 나 [Cur] 중 하나를 선택하여 영역을 설정한 후, 클릭하면 실행된다.
3) 실행이 끝난후 클릭하면 [Generator] 윈도우 화면으로 돌아간다.

설　　명 : 1) ⓔ, ⓕ는 [Converter] 의 ⓒ, ⓓ, ⓔ와 같으므로 그것을 참조하십시오.
2) ⓐ, ⓑ, ⓒ, ⓓ는 각각 2 등분으로 되어있다.
3) ⓐ 왼쪽은 빨간색의 밝기를 표시해주고, 오른쪽은 ⓔ, ⓕ에서 조정한 빨간색의 밝기를 표시한다.
4) ⓑ는 녹색을 ⓐ와 같은 방법으로 표시한다.
5) ⓒ는 파란색을 ⓐ와 같은 방법으로 표시한다.
6) ⓓ의 왼쪽은 ⓐ, ⓑ, ⓒ의 왼쪽을 OR 한 색의 밝기를 표시해 주고, 오른쪽은 ⓔ에서 만들어진 색과 ⓕ에서 만들어진 색을 OR 시켜서 표시해준다.
오른쪽 하단은 ⓕ에서 만들어진 색이 표시된다.
7) ⓖ는 [Effect] 의 ⓑ와 같으므로 그것을 참조하십시오.

[주 의] ⓐ, ⓑ, ⓒ, ⓓ는 다시 지정하지 않는한 그대로 보존된다.

6 수퍼그래픽

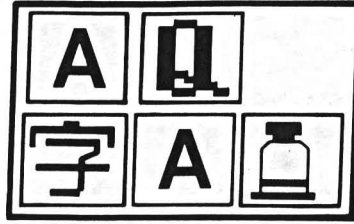
● Digitize (디지타이즈) 는 비디오 화면을 정지된 화면으로 읽어 들이는 기능입니다.
Digitize 를 클릭하면 다음과 같은 윈도우가 표시됩니다.

실행방법 : 1) ⓐ를 설정한다.
2) Start 를 클릭하면, 디지타이즈가 실행된다.
3) 실행을 끝내고 싶으면 2클릭을 한다.

설　명 : 1) ⓐ (페이지) 는 디지타이즈할 화면의 수를 설정하는 것으로 초기치는 1이다.
2) Start 를 클릭하면 윈도우가 사라지고 외부 비디오 입력신호가 디지타이즈 되어 화면에 나타난다. 이때 클릭하면 현재의 디지타이즈 화면이 정지화면으로 되고 다시 클릭하면 계속해서 디지타이즈 화면이 나타난다.
정지화면 상태에서 2클릭하면 Digitize 윈도우로 돌아간다.

● Super on (수퍼 온) 은 컴퓨터의 화면을 수퍼임포즈 상태로 하여 외부 비디오 입력화면과 컴퓨터 화면을 겹쳐 표시할 수 있게 하는 기능입니다. 이 경우 투명색(검은색) 부분만 외부 비디오 화면이 표시됩니다.

● Super off (수퍼 오프) 는 수퍼임포즈 기능을 해제시키는 기능입니다.
Super off 를 클릭하면 Video 윈도우가 사라졌다가 다시 표시되면서 화면이 Video 윈도우 화면으로 돌아갑니다.

10. 字 문자표시

字 아이콘을 클릭하면 다음과 같은 서브 윈도우가 표시됩니다.

● 字 (한자)는 한자를 화면에 표시하는 모드입니다.

실행방법 : 1) 字 를 클릭하면 화면 가운데 정사각형의 커서가 나타나고, 화면의 아래쪽엔 「code input:」이 표시된다.

2) 커서를 커서키로 이동하여 원하는 위치에 놓는다.

3) 원하는 한자의 코드(16진수의 4문자)를 키보드로 입력한다.

4) 지정한 위치에 [빠레트] 에 표시된 색으로 한자가 표시된다.

1. 한자코드가 「0101」부터 있으므로 그 이하의 코드를 지정하면 「UNDEFINED CODE」라는 에러 메시지가 나타난다.
2. 한자코드의 범위를 넘어선 코드를 지정하면, 「ILLEGAL CODE」라는 에러 메시지가 나타난다.
3. 한자나 특수기호, 한글 등이 등록되어 있지 않은 코드를 지정하면 빈공간이 표시된다.

5) 한자지정이 잘못되었을 경우 BS 키를 누르면 바로 전에 표시된 한자가 지워진다. (BS 키를 누를 때마다 실행전 상태와 실행후 상태를 번갈아 나타낸다.)

6) ESC 키를 누르면 한자의 코드입력을 끝내고 字 윈도우가 표시된다.

● A 는 키보드로부터 입력된 문자를 표시하는 모드입니다.

실행방법 : 1) A 를 클릭한 후 원하는 위치에 커서를 놓고 클릭한다.

2) 클릭된 위치에 「—」 표시가 나타난다.

3) 키보드로 문자를 입력하면 「—」가 표시되었던 위치에 문자가 표시되고, 다음 문자를 표시할 위치에 「—」가 표시된다. (표시된 글씨의 색은 [빠레트] 에 표시된 색이다.)

4) 커서키를 누르면 「—」가 지정한 방향으로 한칸씩 이동되어 표시된다.

5) 키보드로부터 문자입력을 끝내려면 커서를 Q 에 놓고 클릭한다. 字 윈도우가 나타난다. (「—」가 표시되어 있지 않으면 실행되지 않는다.)

> **주의** ●화면에서 제일 아래행의 마지막 칸은 문자가 써지지 않습니다.
> ●이 모드에서는 2클릭의 역활이 무효됩니다. (2클릭으로 실행전의 상태로 만들 수 없읍니다.)

● [빠레트] (빠레트)에 표시된 색은 화면에 문자를 표시할 때 나타내는 색입니다.
색의 지정방법은 [도구] 의 [빠레트] 를 참조하십시오.

● RETURN 키를 누르면 커서가 다음 줄의 맨 왼쪽에 표시됩니다.

● 한자코드를 입력하는 방법은 코드표에서 「열」과 「행」의 번호를 입력하면 됩니다.
예를들어 "加"를 표시하고 싶은 경우 「1E03」을 입력하면 됩니다.

● 코드는 자료의 「한자코드(수퍼 그래픽)」을 참조하십시오.

7

자료편

1. 예약어표

예약어는 다음과 같으며, 이것들은 변수명에 포함될 수 없습니다.

ABS
AND
ASC
ATN
ATTR$
AUTO
BASE
BEEP
BIN$
BLOAD
BSAVE
CALL
CDBL
CHR$
CINT
CIRCLE
CLEAR
CLOAD
CLOSE
CLS
CMD
COLOR
CONT
COPY
COS
CSAVE
CSNG
CSRLIN
CVD
CVI
CVS
DATA
DEF
DEFDBL
DEFINT
DEFSNG
DEFSTR
DELETE
DIM
DRAW
DSKF
DSKI$
DSKO$
ELSE
END
EOF
EQV
ERASE
ERL
ERR
ERROR
EXP
FIELD
FILES
FIX
FN
FOR
FPOS
FRE
GET
GO TO
GOSUB
GOTO
HEX$
IF
IMP
INKEY$
INP
INPUT
INSTR
INT
IPL
KEY
KILL
LEFT$
LEN
LET
LFILES
LINE
LIST
LLIST
LOAD
LOC
LOCATE
LOF
LOG
LPOS
LPRINT
LSET
MAX
MERGE
MID$
MKD$
MKI$
MKS$
MOD
MOTOR
NAME
NEW
NEXT
NOT
OCT$
OFF
ON
OPEN
OR
OUT
PAD
PAINT
PDL
PEEK
PLAY
POINT
POKE
POS
PRESET
PRINT
PSET
PUT
READ
REM
RENUM
RESTORE
RESUME
RETURN
RIGHT$
RND
RSET
RUN
SAVE
SCREEN
SET
SGN
SIN
SOUND
SPACE$
SPC
SPRITE
SQR
STEP
STICK
STOP
STR$
STRIG
STRING$
SWAP
TAB
TAN
THEN
TIME
TO
TROFF
TRON
USING
USR
VAL
VARPTR
VDP
VPEEK
VPOKE
WAIT
WIDTH
XOR

2. 문자코드(character code)표

코드 (10진)	코드 (16진)	문자	코드 (10진)	코드 (16진)	문자	코드 (10진)	코드 (16진)	문자	코드 (10진)	코드 (16진)	문자
0	00	제어문자 (CONTROL CHARACTER)	32	20		64	40	@	96	60	ʼ
1	01		33	21	!	65	41	A	97	61	a
2	02		34	22	″	66	42	B	98	62	b
3	03		35	23	#	67	43	C	99	63	c
4	04		36	24	$	68	44	D	100	64	d
5	05		37	25	%	69	45	E	101	65	e
6	06		38	26	&	70	46	F	102	66	f
7	07		39	27	ʼ	71	47	G	103	67	g
8	08		40	28	(	72	48	H	104	68	h
9	09		41	29	)	73	49	I	105	69	i
10	0A		42	2A	*	74	4A	J	106	6A	j
11	0B		43	2B	+	75	4B	K	107	6B	k
12	0C		44	2C	,	76	4C	L	108	6C	l
13	0D		45	2D	—	77	4D	M	109	6D	m
14	0E		46	2E	.	78	4E	N	110	6E	n
15	0F		47	2F	/	79	4F	O	111	6F	o
16	10		48	30	0	80	50	P	112	70	p
17	11		49	31	1	81	51	Q	113	71	q
18	12		50	32	2	82	52	R	114	72	r
19	13		51	33	3	83	53	S	115	73	s
20	14		52	34	4	84	54	T	116	74	t
21	15		53	35	5	85	55	U	117	75	u
22	16		54	36	6	86	56	V	118	76	v
23	17		55	37	7	87	57	W	119	77	w
24	18		56	38	8	88	58	X	120	78	x
25	19		57	39	9	89	59	Y	121	79	y
26	1A		58	3A	:	90	5A	Z	122	7A	z
27	1B		59	3B	;	91	5B	[	123	7B	{
28	1C		60	3C	<	92	5C	₩	124	7C	¦
29	1D		61	3D	=	93	5D	]	125	7D	}
30	1E		62	3E	>	94	5E	∧	126	7E	~
31	1F		63	3F	?	95	5F	_	127	7F	

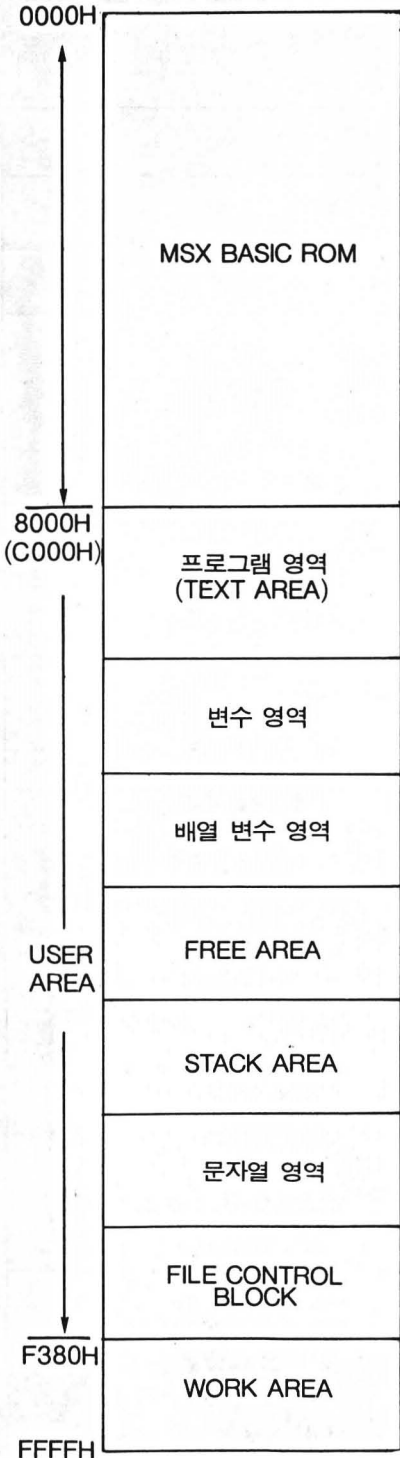
3. 그래픽 문자코드(graphic character code)표

코　　드 (10진)	코　　드 (16진)	문　　자
16	10	π
17	11	┴
18	12	┬
19	13	┤
20	14	├
21	15	┼
22	16	│
23	17	─
24	18	┌
25	19	┐
26	1A	└
27	1B	┘
28	1C	⊠
29	1D	
30	1E	
31	1F	

여기에 있는 그래픽 문자를 키보드로 입력하면 CHR$(1)+CHR$(그래픽 문자코드+64)의 2바이트가 입력됩니다.
또한 PRINT문이나 LPRINT문에 그래픽 문자를 출력하면(예 : PRINT "⊥"), CHR$(1)+CHR$(그래픽 문자코드+64)의 2바이트가 자동적으로 출력됩니다.
여기서 첫문자인 CHR$(1)은 뒤의 문자가 그래픽 문자인 것을 표시하는 제어코드입니다.
그러므로, CHR$ 함수를 사용하여 그래픽 문자를 출력할 경우에는 CHR$(1)+CHR$(그래픽 문자코드+64)로 해야 합니다.

4. 제어코드(control code)표

코드 (10진)	코드 (16진)	기 능	대 응 키
0	00		
1	01	GRAPHIC CHARACTER의 입출력시의 HEADER	CTRL + A
2	02	커서 바로 앞 워드의 선두로 이동	CTRL + B
3	03	입력대기상태를 종료	CTRL + C
4	04		CTRL + D
5	05	커서 이하를 삭제	CTRL + E
6	06	커서를 다음 워드의 맨 앞으로 이동	CTRL + F
7	07	BEEP음을 낸다. (BEEP문과 같읍니다.)	CTRL + G
8	08	커서 바로 앞의 문자를 삭제	CTRL + H, BS
9	09	다음 수평 TAB 위치로 이동	CTRL + I, TAB
10	0A	LINE FEED	CTRL + J
11	0B	커서를 HOME POSITION으로 이동	CTRL + K, HOME
12	0C	화면을 CLEAR하고, 커서를 HOME POSITION으로 이동	CTRL + L, HOME + SHIFT
13	0D	커서를 좌단으로 이동(캐리지 리턴)	CTRL + M, RETURN
14	0E	커서를 행의 끝으로 이동	CTRL + N
15	0F		CTRL + O
16	10		CTRL + P
17	11		CTRL + Q
18	12	삽입모드의 실행 또는 해제	CTRL + R, INS
19	13		CTRL + S
20	14		CTRL + T
21	15	1행을 화면에서 삭제	CTRL + U
22	16		CTRL + V
23	17		CTRL + W
24	18		CTRL + X, SELECT
25	19		CTRL + Y
26	1A		CTRL + Z
27	1B		CTRL + [, ESC
28	1C	커서를 우측으로 이동	CTRL + ₩, →
29	1D	커서를 좌측으로 이동	CTRL +], ←
30	1E	커서를 위로 이동	CTRL + ∧, ↑
31	1F	커서를 아래로 이동	CTRL + SHIFT + −, ↓
127	7F	커서가 있는 위치의 문자가 삭제	DEL

5. 베이직의 메모리 맵

- 행번호를 붙인 프로그램이 저장되는 영역.
- 변수 영역. 문자변수의 경우에는 사용된 문자열을 가르키는 포인터 (string descripter)가 저장된다.
- 배열 변수 영역. 배열변수가 문자형이면 문자열 영역에 사용된 문자열을 가르키는 포인터가 저장된다. 이 영역은 DIM문을 실행한다거나 첨자가 10이하인 배열을 사용할 때에 확보된다.
- 아직 사용하지 않은 영역. 이 영역의 크기는 user area의 크기로부터 문자영역, 스텍 영역, 변수 영역, 프로그램 영역등을 뺀것으로, FRE 함수로서 구할 수 있다.
- FOR～NEXT문이나 GOSUB문등을 실행할 때에 BASIC의 RETURN Address를 보관하기 위한 스텍 영역.
- 문자 변수나 배열 변수에 포함된 문자열을 저장하는 영역.
  문자열 영역의 크기는 CLEAR문으로써 확보된다. 지정하지 않으면 200byte의 영역이 확보된다.
- 화일 입출력에 사용하는 영역.
  MAXFILES문으로써 지정한 수에 대응해서 확보된다.
  상한 어드레스를 CLEAR문으로 F380 이하에 설정할 수 있기 때문에 work area와의 사이에 기계어 루틴등을 위해 사용자가 자유롭게 사용할 수 있는 영역을 설정할 수 있다.
- BASIC이 사용하는 영역

6. 유도함수

다음과 같은 여러가지 함수는 MSX-BASIC에 준비되어 있는 함수를 이용하여 그 값을 구할 수 있읍니다. (오차에 주의하십시오.)

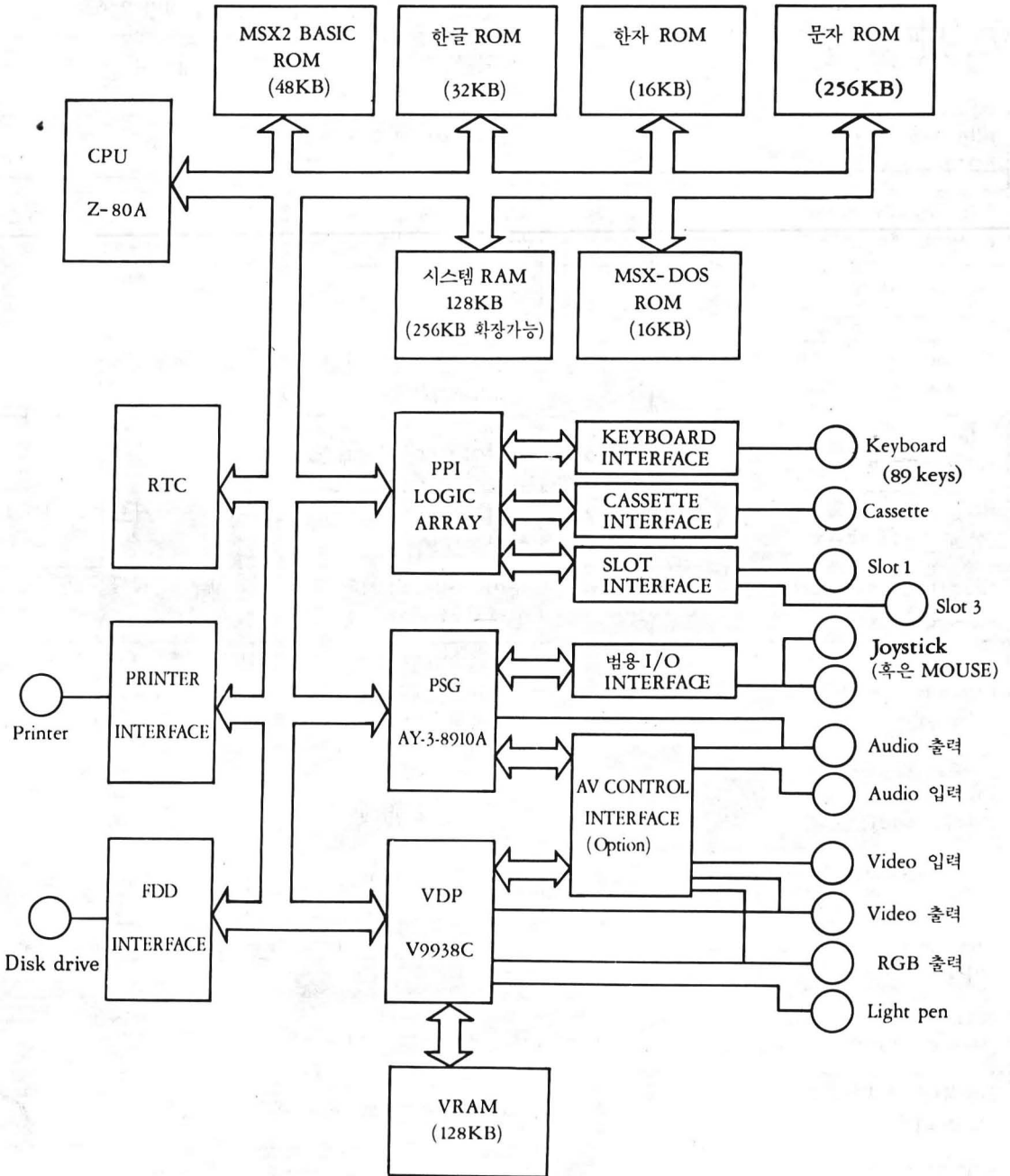
함 수	유 도 식
Secant	SEC(X)=1/COS(X)
Cosecant	CSC(X)=1/SIN(X)
Cotangent	COT(X)=1/TAN(X)
Arc Sine	ARCSIN(X)=ATN(X/SQR(-X*X+1))
Arc Cosine	ARCCOS(X)=-ATN(X/SQR(-X*X+1))+1.5708
Arc Secant	ARCSEC(X)=ATN(SQR(X*X-1))+(SGN(X)-1)*1.5708
Arc Cosecant	ARCCSC(X)=ATN(1/SQR(X*X-1))+(SGN(X)-1)*1.5708
Arc Cotangent	ARCCOT(X)=-ATN(X)+1.5708
Hyperbolic Sine	SINH(X)=(EXP(X)-EXP(-X))/2
Hyperbolic Cosine	COSH(X)=(EXP(X)+EXP(-X))/2
Hyperbolic Tangent	TANH(X)=-EXP(-X)/(EXP(X)+EXP(-X))*2+1
Hyperbolic Secant	SECH(X)=2/(EXP(X)+EXP(-X))
Hyperbolic Cosecant	CSCH(X)=2/(EXP(X)-EXP(-X))
Hyperbolic Cotangent	COTH(X)=EXP(-X)/(EXP(X)-EXP(-X))*2+1
Hyperbolic Arc Sine	ARCSINH(X)=LOG(X+SQR(X*X+1))
Hyperbolic Arc Cosine	ARCCOSH(X)=LOG(X+SQR(X*X-1))
Hyperbolic Arc Tangent	ARCTANH(X)=LOG((1+X)/(1-X))/2
Hyperbolic Arc Secant	ARCSECH(X)=LOG((SQR(-X*X+1)+1)/X)
Hyperbolic Arc Cosecant	ARCCSCH(X)=LOG((SGN(X)*SQR(X*X+1)+1)/X)
Hyperbolic Cotangent	ARCCOTH(X)=LOG((X+1)/(X-1))/2

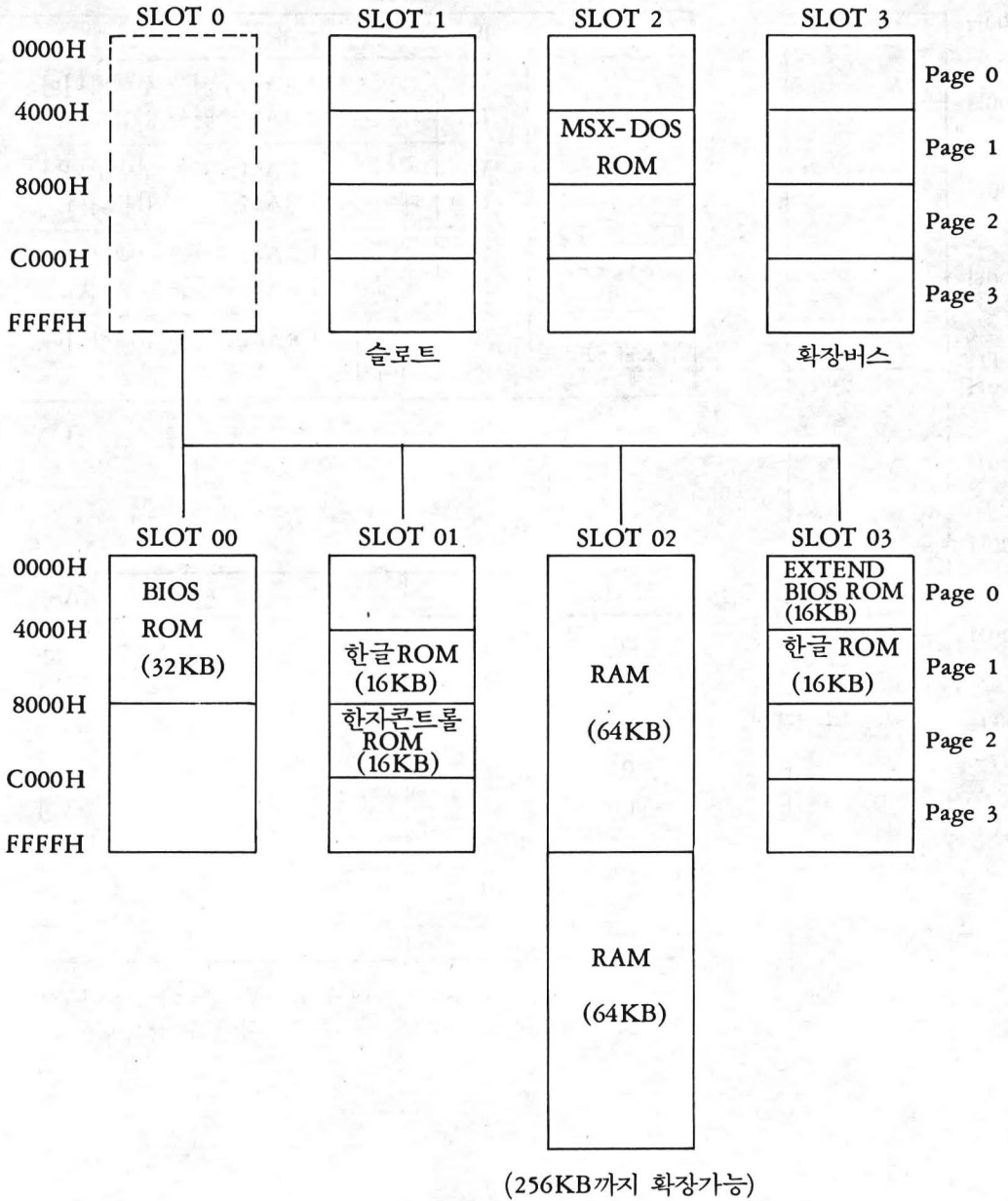
7. 에러 메세지(error message)

메 세 지	에러코드	설 명
Bad drive name (배드 드라이브 네임)	62	사용하지 않는 디스크 드라이브의 이름을 디바이스 명으로 지정하였을 경우
Bad FAT (배드 에프에이티)	60	디스켓이 FORMAT되지 않았을 경우
Bad file mode (배드 화일 모드)	61	OPEN문에서 「FOR INPUT」, 「FOR OUTPUT」, 「FOR APPEND」 또는 아무것도 지정하지 않았을 경우 등을 제외한 화일모드를 지정하였을 경우, 시퀀셜 화일에 대해서 PUT, GET, LOF를 사용했을 경우, 또는 랜덤엑세스 화일에 대해 LOAD 명령문이 실행되었을 경우에 나타납니다.
Bad file name (배드 화일 네임)	56	화일명이 적당하지 않을 경우
Bad file number (배드 화일 넘버)	52	열려있지 않은 화일번호를 지정할 경우. 지정한 화일번호가 MAXFILES문에서 지정한 수를 초월할 경우
Bad sector number (배드 섹터 넘버)	63	PUT, GET 명령문으로 지정한 레코드번호가 0～32767을 벗어났을 경우
Can't CONTINUE (캔트 콘티뉴)	17	CONT 명령으로 프로그램의 실행을 다시 개시하지 않습니다. 에러가 발생하여 종료된 프로그램은 다시 개시할 수 없읍니다. 또한 프로그램의 일부를 바꿔 쓴다거나, CLEAR문 등을 실행하면 다음에 실행할 포인터(POINTER)가 파괴되어 프로그램을 다시 개시할 수 없읍니다.
Direct Statement in file (다이렉트 스테이트먼트 인 화일)	19	데이타 레코더, 프린터, CRT 등으로 입출력하는 동안 에러가 발생할 경우
Direct Statement (다이렉트 스테이트먼트)	57	ASCII 형식의 프로그램 화일을 LOAD하는 동안, 프로그램 이외의 데이타가 있을 경우
Disk full (디스크 풀)	66	디스켓에 남은 용량이 없을 경우
Disk I/O error (디스크 아이오 에러)	69	디스켓과의 입출력시에 회복 불가능한 에러가 발생했을 경우
Disk offline (디스크 오프 라인)	70	지정된 드라이브에 디스켓이 들어있지 않을 경우
Disk write protected (디스크 라이트 프로텍티드)	68	WRITE가 금지된 디스켓에 PUT 또는 PRINT #문 등을 실행하였을 경우

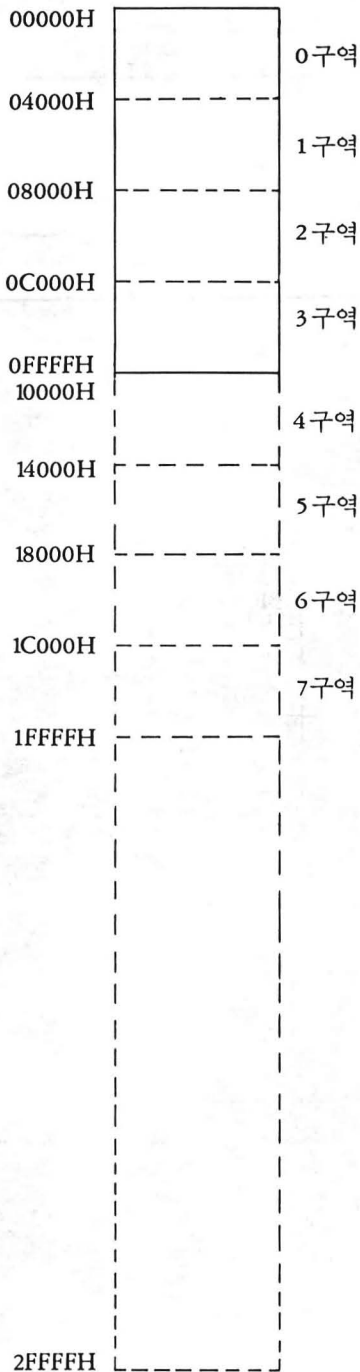
메 세 지	에러코드	설 명
Out of string space (아웃 오브 스트링 스페이스)	14	문자열 변수용의 메모리가 부족한 경우
Overflow (오버플로우)	6	연산결과나 입력된 수치가 형에 대응한 허용범위를 넘는 경우
Redimensioned array (리디멘젼 어레이)	10	배열변수를 이중으로 정의하는 경우. 배열변수에 첨자의 최대치 10이 할당된 후에 배열을 정의하는 경우
Rename across disk (리네임 어크로스 디스크)	71	서로 다른 디스크 드라이브 사이에서 NAME 문을 실행했을 경우
RESUME without error (리슘 윋아웃 에러)	22	에러처리 루틴 이외에서 RESUME문을 사용하는 경우
RETURN without GOSUB (리턴 윋아웃 고서브)	3	GOSUB문과 RETURN문이 대응하고 있지 않을 경우
Sequential I/O only (시퀀셜 아이오 온리)	58	Sequential 화일에 대하여 랜덤 입출력을 행하는 경우. 이 에러메세지는 장래의 확장에 대하여 준비되 있는 것입니다.
String formula too complex (스트링 포뮬라 투 컴플렉스)	16	지정한 문자열이 복잡할 경우. 문자식을 두개 이상으로 나누십시오.
String too long (스트링 투 롱)	15	문자열이 255문자보다 클 경우
Subscript out of range (서브스크립트 아웃 오브 레인지)	9	지정한 배열변수의 첨자의 값이 DIM문에서 지정한 배열변수의 첨자의 최대치보다 클 경우
Syntax error (신택스 에러)	2	문을 쓰는 방법이 틀렸을 경우(괄호가 대응하지 않고 명령, 문의 조작이 틀려있고, 구절의 기호가 없는등)
Too many files (투 매니 화일즈)	67	화일의 수가 112개를 초과했을 경우
Type mismatch (타잎 미스매치)	13	수식이나 수치 함수에 문자열을 사용하고, 문자식이나 문자 함수에 수치를 사용하는 등, 데이타의 형식이 일치하지 않을 경우
Undefined line number (언디파인드 라인 넘버)	8	GOTO문, GOSUB문, IF문 등에서 지정한 행번호가 프로그램중에 없을 경우
Undefined user function (언디파인드 유져 펑션)	18	DEF FN으로 정의하지 않은 USER 정의함수를 사용하고 있을 경우
Unprintable error (언프린터블 에러)	23 26～49 72～255	Message가 정의되지 않은 에러
Verify error (베리파이 에러)	20	카세트에 저장된 프로그램이 메모리상의 프로그램과 내용이 다를 경우 프로그램의 저장을 바르게 하십시오.

메 세 지	에러코드	설 명
Division by zero (디비젼 바이 제로)	11	0으로 나눌때 혹은 0에 대하여 음의 멱승을 취할 경우
FIELD overflow (필드 오버플로우)	50	화일명에서 정의한 영역 크기의 합계가 256 바이트를 초월하는 경우 이 에러 메세지는 장래의 확장에 대하여 준비된 것입니다.
File already exists (화일 올레디 이그지스트)	65	NAME명령으로 지정한 새로운 화일명이 이미 디스켓에 존재하고 있을 경우
File already open (화일 올레디 오픈)	54	OPEN문에서 지정한 화일번호를 다른 화일에서 사용할 경우
File not found (화일 낫 파운드)	53	지정된 화일이 없을 경우
File not open (화일 낫 오픈)	59	OPEN문에서 열려있지 않은 화일에 대하여 입출력을 행하는 경우
File still open (화일 스틸 오픈)	64	화일이 CLOSE되어 있지 않을 경우
Illegal direct (일레갈 다이렉트)	12	직접모드에서 실행될 수 없는 문을 직접모드에서 실행하는 경우
Illegal function call (일레갈 펑션 콜)	5	문이나 함수의 사용법이 틀린 경우 인수등이 지정 가능한 범위를 초월했을 경우
Input past end (인풋 패스트 앤드)	55	화일중의 데이타를 모두 읽어들인 후, 다시 입력문을 실행하는 경우 화일중의 데이타를 모두 읽어 들였는가 어떤가는 EOF 함수로 점검하십시오.
Internal error (인터널 에러)	51	베이직 내부에 에러가 발생하는 경우
Line buffer overflow (라인 버퍼 오버플로우)	25	입력된 데이타가 허용하는 문자수를 초월한 경우
Missing operand (미싱 오퍼랜드)	24	문중에 필요한 파라메터가 지정되어 있지 않는 경우
NEXT without FOR (넥스트 윋아웃 포)	1	NEXT문에 대응하는 FOR문이 없을 경우
No RESUME (노우 리슘)	21	에러 처리루틴에 RESUME 문이 없을 경우
Out of DATA (아웃 오브 데이타)	4	READ문에서 읽어낼 데이타가 없을 경우
Out of memory (아웃 오브 메모리)	7	프로그램이 길거나 배열이 크게 되어 메모리가 부족할 경우

8. 시스템 블록 다이어그램(System block diagram)

9. 슬로트 할당(slot management)

(256KB까지 확장가능)

10. 메모리 맵퍼(memory mapper)

RAM ADDRESS	RAM구역
00000H ~ 04000H	0구역
04000H ~ 08000H	1구역
08000H ~ 0C000H	2구역
0C000H ~ 0FFFFH	3구역
10000H ~ 14000H	4구역
14000H ~ 18000H	5구역
18000H ~ 1C000H	6구역
1C000H ~ 1FFFFH	7구역
1FFFFH ~ 2FFFFH	

1. Memory Mapper를 위한 I/O ADDRESS 할당

I/O ADDR	RW	내 용
&H FC	W R	페이지 0의 RAM구역을 선택합니다. 페이지 0의 RAM구역을 읽읍니다.
&H FD	W R	페이지 1의 RAM구역을 선택합니다. 페이지 1의 RAM구역을 읽읍니다.
&H FE	W R	페이지 2의 RAM구역을 선택합니다. 페이지 2의 RAM구역을 읽읍니다.
&H FF	W R	페이지 3의 RAM구역을 선택합니다. 페이지 3의 RAM구역을 읽읍니다.

2. RAM 구역 선택방법

MA17	MA16	MA15	MA14	RAM 구역
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7

* 페이지 3의 RAM구역을 컴퓨터가 동작중인 상태에서 바꾸면 시스템이 오동작하는 수도 있으니 주의하여 주십시오.

※ X-II에서는 시스템 메모리가 128KB이지만 메모리를 확장하여 최대 256KB까지 사용할 수 있읍니다.

11. I/O 맵(input/output map)

1. I/O 어드레스 할당

I/O 어드레스	입출력	내　　　　용	비　고
ØØH～3FH		지정되어 있지 않음	
4ØH～7FH		예　약	
*8ØH～87H		RS-232C	
88H～8BH		예　약	
*8CH～8DH		Modem	
8EH～8FH		예　약	
9ØH～91H 9ØH 91H	 R/W W	Printer bit Ø: Strobe output bit 1: Status read data write	
92H～97H		예　약	
98H～9BH 98H 99H 9AH 9BH	 R/W R/W W W	VDP VRAM access Command register access, address set status register read Palette register access Index register pointer	V9938C
9CH～9FH		예　약	
AØH～A3H AØH A1H A2H	 W W R	PSG address latch data write data read	AY-3-8910A
A4H～A7H		예　약	
A8H～ABH A8H A9H AAH ABH	 R/W R/W R/W W	Parallel port Port A Port B Port C mode Set	DW64M×1
ACH～AFH		예　약	
BØH～B3H		예　약	

I/O 어드레스	입 출 력	내 용	비 고
B4H～B5H B4H B5H	 W R/W	Clock IC address latch data read/write	RP5C01
B6H～B7H		예 약	
B8H～BBH		예 약	
BCH～BFH		예 약	
CØH～C1H		예 약	
C2H～C7H		예 약	
C8H～CFH		예 약	
*DØH～D7H		FDC	
D8H～D9H D8H D9H	 W W R	한자 ROM bØ～b5 하위 address set bØ～b6 상위 address set data read	
DAH～DBH		예 약	
DCH～F4H		예 약	
F5H		예 약	
F6H		예 약	
*F7H	 W W W R/W W W W W	A/V Control bØ: audio R("Ø"이면 mixing on) b1: audio L("Ø"이면 mixing on) b2: video input select("Ø"이면 RGB) b3: 외부 sync control("Ø"이면 내부동기) b4: A/V control("Ø"이면 A/V off) b5: intensity control("Ø"이면 full intensity) b6: mode control b7: mode control	
F8H～FBH		예 약	
FCH～FFH FCH FDH FEH FFH	 R/W R/W R/W R/W	Memory mapper 페이지Ø의 RAM 구역 선택 페이지 1의 RAM 구역 선택 페이지 2의 RAM 구역 선택 페이지 3의 RAM 구역 선택	 DW 64M×1

*로 표시된 것은 옵션(Option) 기기를 위하여 준비되어 있다는 것을 의미합니다.

2.PPI bit 할당

<table>
<tr><th>port</th><th>bit</th><th>I/O</th><th>신 호 명</th><th>내 용</th></tr>
<tr><td rowspan="4">A</td><td>0
1</td><td rowspan="4">출
력</td><td>CS0L
CS0H</td><td>0000～3FFF 번지의 Slot 지정 번호</td></tr>
<tr><td>2
3</td><td>CS1L
CS1H</td><td>4000～7FFF 번지의 Slot 지정 번호</td></tr>
<tr><td>4
5</td><td>CS2L
CS2H</td><td>8000～BFFF 번지의 Slot 지정 번호</td></tr>
<tr><td>6
7</td><td>CS3L
CS3H</td><td>C000～FFFF 번지의 Slot 지정 번호</td></tr>
<tr><td>B</td><td>0
≀
7</td><td>입
력</td><td></td><td>Keyboard Return 신호</td></tr>
<tr><td rowspan="5">C</td><td>0
1
2
3</td><td rowspan="5">출
력</td><td>KB0
KB1
KB2
KB3</td><td>Keyboard Scan 신호</td></tr>
<tr><td>4</td><td>CASON</td><td>카세트 Control(L-ON)</td></tr>
<tr><td>5</td><td>CASW</td><td>카세트에 써 넣는 신호</td></tr>
<tr><td>6</td><td>CAPS</td><td>CAPS Lamp 신호(L로 점등)</td></tr>
<tr><td>7</td><td>SOUND</td><td>Soft ware에 의한 Sound 출력</td></tr>
</table>

3. PSG bit 할당

port	bit	I/O	접속 Connector된 번호	Joystick 사용시의 신호
A	0	입력	J3-1핀 *1	FWD1
			J4-1핀 *2	FWD2
	1		J3-2핀 *1	BACK1
			J4-2핀 *2	BACK2
	2		J3-3핀 *1	LEFT1
			J4-3핀 *2	LEFT2
	3		J3-4핀 *1	RIGHT1
			J4-4핀 *2	RIGHT2
	4		J3-6핀 *1	TRGA1
			J4-6핀 *2	TRGA2
	5		J3-7핀 *1	TRGB1
			J4-7핀 *2	TRGB2
	6		사용되지 않음	"H" 레벨
	7		CASR (카세트 테이프의 Read)	
B	0	출력	J3-6핀 *3	"H" 레벨
	1		J3-7핀 *3	"H" 레벨
	2		J4-6핀 *3	"H" 레벨
	3		J4-7핀 *3	"H" 레벨
	4		J3-8핀	
	5		J4-8핀	
	6		port A 입력 select	
	7		KLAMP	한글램프(L로 점등)

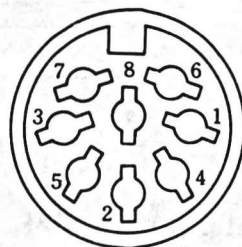
*1. port B의 bit 6이 "L" Level시 유효. Joystick 1용.
*2. port B의 bit 6이 "H" Level시 유효. Joystick 2용.
*3. 출력 port로서 사용되지 않을시는 "H" Level로 되지 않습니다.

12. 접속단자

카세트 접속단자

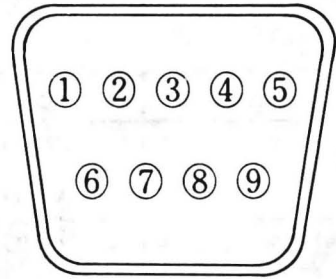
○입 력 : 카세트의 이어폰 단자에 접속
○출 력 : 카세트의 마이크로폰 단자에 접속
○동 기 방 식 : 비 동기방식
○전 송 속 도 : 1200 baud(1200Hz—1파 "0", 2400Hz—2파 "1")
2400 baud(2400Hz—1파 "0", 4800Hz—2파 "1")
○변 조 방 식 : FSK 방식
○리모트기능 : 가능
○콘 넥 터 : DIN 8핀
○신 호 선 표

단자번호	신 호 명
1	GND
2	GND
3	GND
4	CMTOUT
5	CMTIN
6	REM+
7	REM−
8	GND

핀 Connection(코넥숀)

조이스틱 접속단자

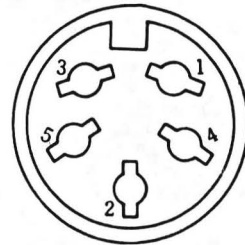
○ 입출력 : 입력 4bit, 출력 1bit, 쌍방향 2bit
○ Logic : 정논리
○ 레 벨 : TTL
○ 신호선표

단자번호	신 호 명
1	FWD
2	BACK
3	LEFT
4	RIGHT
5	+5V
6	TRG1
7	TRG2
8	출력
9	GND

핀 Connection (코넥숀)

전자펜(Light pen) 접속단자

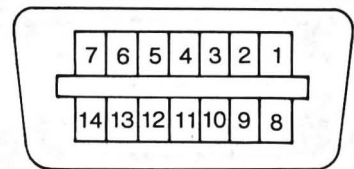
○ 신호선표

단자번호	신 호 명
1	+5V
2	AMP GAIN
3	GND
4	전자펜 detect
5	전자펜 스위치

핀 Connection (코넥숀)

Printer (프린터) 접속단자

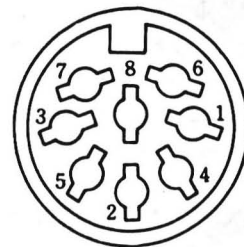
○Logic : 정논리
○Level : TTL
○신호선표

단자번호	방향	설 명
1	출력	PST (스트로브) 신호
2	출력	PDB 0 프린트데이타 비트 0
3	출력	PDB 1 프린트데이타 비트 1
4	출력	PDB 2 프린트데이타 비트 2
5	출력	PDB 3 프린트데이타 비트 3
6	출력	PDB 4 프린트데이타 비트 4
7	출력	PDB 5 프린트데이타 비트 5
8	출력	PDB 6 프린트데이타 비트 6
9	출력	PDB 7 프린트데이타 비트 7
10	—	
11	입력	BUSY (비지) 신호
12	—	
13	—	
14	—	GND 전원 0V

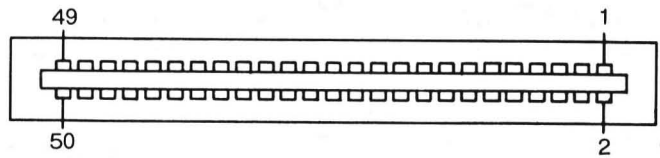
핀 Connection (코넥숀)

RGB 영상신호 단자

○신호선표

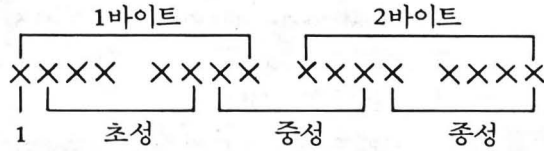
단자번호	신 호 명
1	AUDIO
2	GND
3	Composite Sync
4	Fast Blanking
5	+12V
6	RED
7	GREEN
8	BLUE

핀 Connection (코넥숀)

13. MSX 버스

핀번호	명 칭	I/O	핀번호	명 칭	I/O
1	$\overline{\text{CS1}}$	O	2	$\overline{\text{CS2}}$	O
3	$\overline{\text{CS12}}$	O	4	$\overline{\text{SLTSL}}$	O
5	예비 주(1)	—	6	$\overline{\text{RFSH}}$	O
7	$\overline{\text{WAIT}}$ 주(2)	I	8	$\overline{\text{INT}}$ 주(2)	I
9	$\overline{\text{M1}}$	O	10	$\overline{\text{BUSDIR}}$	I
11	$\overline{\text{IORQ}}$	O	12	$\overline{\text{MERQ}}$	O
13	$\overline{\text{WR}}$	O	14	$\overline{\text{RD}}$	O
15	$\overline{\text{RESET}}$	O	16	예비 주(1)	—
17	A9	O	18	A15	O
19	A11	O	20	A10	O
21	A7	O	22	A6	O
23	A12	O	24	A8	O
25	A14	O	26	A13	O
27	A1	O	28	A0	O
29	A3	O	30	A2	O
31	A5	O	32	A4	O
33	D1	I/O	34	D0	I/O
35	D3	I/O	36	D2	I/O
37	D5	I/O	38	D4	I/O
39	D7	I/O	40	D6	I/O
41	GND	—	42	CLOCK	O
43	GND	—	44	SW1	—
45	+5V	—	46	SW2	—
47	+5V	—	48	+12V	—
49	SOUNDIN	I	50	−12V	—

주(1) 예비는 사용금지단자.
주(2) OPEN COLLECTOR입력

핀번호	명 칭	내 용
1	$\overline{\text{CS1}}$	ROM 4000～7FFF번지 Select 신호
2	$\overline{\text{CS2}}$	ROM 8000～BFFF번지 Select 신호
3	$\overline{\text{CS12}}$	ROM 4000～BFFF번지 Select 신호
4	$\overline{\text{SLTSL}}$	Slot Select 신호. 각 Slot 단자에는 그 Slot 고유의 Select 신호가 부가된다.
5	예비	장래를 위한 예비 신호선. 사용금지
6	$\overline{\text{RFSH}}$	refresh Cycle 신호
7	$\overline{\text{WAIT}}$	CPU의 Wait 요구 신호
8	$\overline{\text{INT}}$	CPU의 Interrupt 요구신호
9	$\overline{\text{M1}}$	CPU의 Fetch Cycle을 표시하는 신호
10	$\overline{\text{BUSDIR}}$	외부 Data Bus Buffer의 방향을 제어하는 신호
11	$\overline{\text{IORQ}}$	I/O Request 신호
12	$\overline{\text{MERQ}}$	Memory Request 신호
13	$\overline{\text{WR}}$	Write Timming 신호
14	$\overline{\text{RD}}$	Read Timming 신호
15	$\overline{\text{RESET}}$	시스템 Reset 신호
16	예비	장래를 위한 예비 신호선. 사용금지
17～32	A0～A15	Address Bus
33～40	D0～D7	Data Bus
41	GND	Ground
42	Clock	CPU Clock (3.579545 MHz)
43	GND	Ground
44, 46	SW1, SW2	Protect용 Switch
45, 47	+5V	+5V 전원
48	+12V	+12V 전원
49	SOUNDIN	Sound 입력 신호
50	−12V	−12V 전원

14-1. 한글코드표(2바이트 분리형)

2바이트 분리형 한글코드는 초성, 중성, 종성이 각각 5비트로 구성되어 있으며, 맨 앞의 한 비트는 한글식별비트로, 항상 1로 설정되어 있읍니다.

비트 54321	초 성	중 성	종 성
00000	not used	restricted	restricted
00001	fill code	restricted	fill code
00010	ㄱ	fill code	ㄱ
00011	ㄲ	ㅏ	ㄲ
00100	ㄴ	ㅐ	ㄳ
00101	ㄷ	ㅑ	ㄴ
00110	ㄸ	ㅒ	ㄵ
00111	ㄹ	ㅓ	ㄶ
01000	ㅁ	restricted	ㄷ
01001	ㅂ	restricted	ㄹ
01010	ㅃ	ㅔ	ㄺ
01011	ㅅ	ㅕ	ㄻ
01100	ㅆ	ㅖ	ㄼ
01101	ㅇ	ㅗ	ㄽ
01110	ㅈ	ㅘ	ㄾ
01111	ㅉ	ㅙ	ㄿ
10000	ㅊ	restricted	ㅀ
10001	ㅋ	restricted	ㅁ
10010	ㅌ	ㅚ	not used
10011	ㅍ	ㅛ	ㅂ
10100	ㅎ	ㅜ	ㅄ
10101	not used	ㅝ	ㅅ
10110	not used	ㅞ	ㅆ
10111	not used	ㅟ	ㅇ
11000	not used	restricted	ㅈ
11001	not used	restricted	ㅊ
11010	not used	ㅠ	ㅋ
11011	not used	ㅡ	ㅌ
11100	not used	ㅢ	ㅍ
11101	not used	ㅣ	ㅎ
11110	not used	not used	not used
11111	restricted	not used	restricted

14-2. 한글코드표(N바이트형)

	상위 4 비트 (16진수)															
하위 4 비트 (16진수)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		π		0	@	P	`	p	♠	ㅆ	ㅖ	끄	로	뾰	우	츄
1		┴	!	1	A	Q	a	q	♥	ㅇ	ㅗ	노	료	뿌	유	츠
2	포	┬	”	2	B	R	b	r	♣	ㅈ	ㅛ	뇨	루	쀼	으	코
3	표	┤	#	3	C	S	c	s	◆	ㅉ	ㅜ	누	류	쁘	조	쿄
4	푸	├	$	4	D	T	d	t	○	ㅊ	ㅠ	뉴	르	소	죠	쿠
5	퓨	┼	%	5	E	U	e	u	●	ㅋ	ㅡ	느	모	쇼	주	큐
6	프	│	&	6	F	V	f	v	ㄱ	ㅌ	ㅣ	도	묘	수	쥬	크
7	호	─	’	7	G	W	g	w	ㄲ	ㅍ	고	됴	무	슈	즈	토
8	효	┌	(	8	H	X	h	x	ㄴ	ㅎ	교	두	뮤	스	쪼	툐
9	후	┐	)	9	I	Y	i	y	ㄷ	ㅏ	구	듀	므	쏘	쬬	투
A	휴	└	*	:	J	Z	j	z	ㄸ	ㅐ	규	드	보	쑈	쭈	튜
B	흐	┘	+	;	K	[	k	{	ㄹ	ㅑ	그	또	뵤	쑤	쮸	트
C	│	╳	,	<	L	₩	l	¦	ㅁ	ㅒ	꼬	뚀	부	쓔	쯔	
D	▕		—	=	M	]	m	}	ㅂ	ㅓ	꾜	뚜	뷰	쓰	초	
E	‖		·	>	N	∧	n	~	ㅃ	ㅔ	꾸	뜌	브	오	쵸	
F	‖		/	?	O	—	o		ㅅ	ㅕ	뀨	뜨	뽀	요	추	

주의 한글코드 가운데 "고", "교"~"흐"에 해당하는 코드는 단지 화면표시를 위한 것이며, 메모리상에는 "ㄱ", "ㄴ", "~", "ㅣ"에 해당하는 자소 단위로 저장됩니다.
따라서 한글코드중에 키보드상에서 의미가 있는 코드는 &H86("ㄱ")부터 &HA6("ㅣ")까지입니다.

7 자료편

15. 한자코드표(시스템)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D44X	·	假	價	加	可	家	暇	架	歌	街	刻	却	各	脚	覺	角
		가									각					
D45X	閣	刊	干	幹	看	簡	肝	間	渴	葛	感	敢	減	甘	監	·
		간							갈		감					
D46X	·	鑑	甲	剛	姜	康	強	江	降	綱	講	鋼	介	個	改	槪
			갑	강									개			
D47X	皆	蓋	開	客	更	去	居	巨	拒	據	擧	距	乾	件	健	·
				객	갱	거							건			
D48X	·	建	傑	儉	劍	檢	擊	格	激	隔	堅	絹	見	遣	決	潔
			걸	검			격				견				결	
D49X	結	缺	訣	兼	謙	京	傾	卿	境	庚	徑	慶	敬	景	更	·
				겸		경										
D4AX	·	硬	競	經	耕	警	鏡	頃	輕	係	啓	契	季	戒	桂	械
										계						
D4BX	溪	界	癸	系	繼	計	階	鷄	古	告	固	孤	庫	故	稿	·
									고							
D4CX	·	考	苦	顧	高	鼓	曲	穀	谷	困	坤	骨	供	公	共	功
							곡			곤		골	공			
D4DX	孔	工	恐	攻	空	貢	寡	果	科	課	過	郭	冠	官	寬	·
							과					곽	관			
D4EX	·	款	管	觀	關	館	括	光	廣	狂	鑛	掛	傀	塊	壞	怪
							괄	광				괘	괴			
D4FX	交	巧	敎	校	橋	矯	較	郊	丘	久	九	具	區	口	句	·
	교								구							

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D54X	·	拘	救	構	歐	求	球	究	舊	購	邱	龜	國	局	菊	君
		구											국			군
D55X	群	軍	郡	宮	弓	券	卷	圈	拳	權	軌	歸	貴	龜	圭	·
				궁		권					궤	귀			규	
D56X	·	奎	規	均	菌	克	劇	極	勤	根	近	今	琴	禁	金	錦
				균		극			근			금				
D57X	及	急	級	給	肯	企	其	器	基	奇	寄	岐	己	忌	技	·
	급				긍	기										
D58X	·	旗	既	期	棋	棄	機	欺	氣	畿	祈	紀	記	起	騎	緊
																긴
D59X	吉	金	那	諾	暖	難	南	男	納	乃	内	耐	女	年	念	·
	길	김	나	낙	난		남		납	내			녀	년	념	
D5AX	·	寧	努	奴	濃	農	腦	尿	能	泥	多	茶	丹	但	單	團
		녕	노		농		뇌	뇨	능	니	다		단			
D5BX	壇	斷	旦	檀	段	短	端	達	擔	淡	潭	談	答	踏	唐	·
								달	담				답		당	
D5CX	·	堂	當	糖	黨	代	垈	大	對	帶	待	臺	貸	隊	德	悳
						대									덕	
D5DX	倒	刀	到	圖	導	島	度	徒	挑	渡	盜	途	道	都	陶	·
	도															

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D64X	·	毒	獨	督	讀	敦	豚	突	冬	凍	動	同	東	洞	童	銅
		독				돈		돌	동							
D65X	斗	杜	豆	頭	鈍	得	燈	登	等	騰	羅	樂	洛	絡	落	·
	두				둔	득	등				라	락				
D66X	·	亂	卵	蘭	覽	廊	浪	郞	朗	來	冷	略	兩	梁	涼	糧
		란			람	랑				래	랭	략	량			
D67X	良	諒	量	呂	慮	旅	麗	力	歷	曆	戀	練	聯	蓮	連	·
				려				력			련					
D68X	·	鍊	列	烈	令	嶺	零	靈	領	例	禮	勞	爐	盧	老	路
			렬		령					례		로				
D69X	露	魯	鷺	祿	綠	錄	論	賴	了	僚	料	龍	累	劉	柳	·
				록			론	뢰	료			룡	루	류		
D6AX	·	流	留	類	六	陸	倫	輪	律	栗	率	隆	陵	利	吏	履
					륙		륜		률			륭	릉	리		
D6BX	李	梨	理	里	離	隣	林	臨	立	粒	摩	磨	馬	麻	幕	·
						린	림		립		마				막	
D6CX	·	慢	晩	滿	灣	萬	末	亡	忘	望	網	埋	妹	媒	梅	每
		만					말	망				매				
D6DX	買	賣	脈	麥	孟	盟	盲	免	眠	綿	面	滅	名	命	明	·
			맥		맹			면				멸	명			
D6EX	·	銘	鳴	募	暮	某	模	母	毛	謀	謨	貌	木	沐	牧	目
				모									목			
D6FX	睦	夢	卯	墓	妙	苗	務	戊	武	無	舞	茂	貿	霧	墨	·
		몽	묘				무								묵	

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D74X	·	默	問	文	紋	聞	門	物	味	尾	微	未	米	美	迷	敏
		묵	문					물	미							민
D75X	民	閔	密	博	朴	舶	迫	半	反	叛	班	盤	般	返	飯	·
			밀	박				반								
D76X	·	拔	發	髮	傍	房	放	方	紡	芳	訪	邦	防	倍	北	培
		발			방									배		
D77X	拜	排	抔	背	裵	配	伯	白	百	番	繁	翻	罰	凡	汎	·
							백			번			벌	범		
D78X	·	犯	範	法	壁	碧	卞	變	辯	邊	別	丙	倂	兵	炳	病
				법	벽		변				별	병				
D79X	秉	保	報	寶	普	步	甫	補	譜	伏	ト	復	服	福	腹	·
		보								복						
D7AX	·	複	覆	本	俸	奉	封	峰	縫	蜂	鋒	鳳	付	副	否	夫
				본	봉								부			
D7BX	婦	富	府	復	扶	浮	父	符	簿	腐	膚	負	賦	部	釜	·
D7CX	·	附	北	分	墳	憤	粉	紛	不	佛	弗	拂	朋	備	妃	悲
			북	분					불				붕	비		
D7DX	批	比	碑	秘	肥	費	非	飛	鼻	貧	賓	頻	氷	事	仕	·
										빈			빙	사		

7 자료편

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D84X	·	使	司	史	四	士	寫	寺	射	師	思	斜	斯	査	死	沙
		사														
D85X	祀	社	私	絲	舍	詐	詞	謝	辭	飼	削	山	散	産	算	·
											삭	산				
D86X	·	酸	殺	三	參	森	蔘	上	傷	像	償	商	喪	尚	常	床
			살	삼				상								
D87X	想	桑	狀	相	祥	裳	詳	象	賞	霜	塞	色	索	生	省	·
											새	색		생		
D88X	·	序	庶	徐	敍	暑	書	瑞	緖	署	西	夕	席	析	石	碩
		서										석				
D89X	釋	錫	仙	先	善	宣	旋	禪	線	船	選	鮮	舌	設	說	·
			선										설			
D8AX	·	雪	纖	涉	燮	城	姓	性	成	星	盛	省	聖	聲	誠	世
			섬	섭		성										세
D8BX	勢	歲	洗	稅	細	貰	召	小	少	巢	所	昭	消	燒	笑	·
							소									
D8CX	·	素	蘇	訴	騷	俗	屬	束	續	速	孫	損	率	宋	松	訟
						속					손		솔	송		
D8DX	送	頌	刷	殺	修	受	垂	壽	守	愁	手	授	收	數	樹	·
			쇄		수											
D8EX	·	殊	水	洙	秀	輸	遂	隨	需	須	首	宿	淑	熟	肅	巡
												숙				순
D8FX	淳	純	舜	順	戌	術	述	崇	習	襲	乘	僧	勝	升	承	·
					술			숭	습		승					

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D94X	·	昇	始	市	施	是	時	示	視	詩	試	式	息	植	殖	識
		승	시									식				
D95X	食	飾	伸	信	愼	新	申	神	腎	臣	身	辛	失	室	實	·
			신										실			
D96X	·	審	心	沈	深	甚	十	拾	雙	氏	亞	兒	我	牙	阿	岳
		심					십		쌍	씨	아					악
D97X	嶽	惡	握	安	岸	案	眼	顔	巖	暗	癌	壓	押	鴨	仰	·
				안					암			압			앙	
D98X	·	央	哀	愛	液	額	夜	野	弱	約	若	藥	壤	揚	楊	様
			애		액		야		약				양			
D99X	洋	羊	讓	陽	養	御	於	漁	語	魚	億	抑	彦	言	嚴	·
						어					억		언		엄	
D9AX	·	業	余	如	汝	與	餘	域	役	易	疫	譯	逆	驛	宴	延
		업	여					역							연	
D9BX	沿	淵	演	煙	然	燃	燕	研	縁	軟	鉛	悦	熱	染	炎	·
												열		염		

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DA4X	·	鹽	葉	影	映	榮	永	泳	營	英	迎	藝	譽	豫	銳	預
		염	엽	영								예				
DA5X	五	午	吳	娛	梧	汚	烏	誤	屋	沃	獄	玉	溫	翁	瓦	·
	오								옥				온	옹	와	
DA6X	·	臥	完	緩	往	旺	王	倭	外	搖	曜	腰	要	謠	慾	欲
			완		왕			왜	외	요					욕	
DA7X	浴	勇	容	庸	溶	用	踊	鎔	鏞	于	偶	優	友	右	宇	·
		용								우						
DA8X	·	愚	憂	牛	祐	禹	羽	遇	郵	雨	旭	云	運	雲	韻	熊
											욱	운				웅
DA9X	雄	元	原	員	圓	園	怨	援	源	苑	遠	院	願	月	越	·
		원												월		
DAAX	·	位	偉	僞	危	圍	委	威	慰	爲	緯	胃	衛	違	乳	儒
		위													유	
DABX	兪	幼	幽	惟	有	柔	油	由	維	裕	誘	遊	遺	酉	肉	·
															육	
DACX	·	育	允	尹	潤	融	恩	銀	隱	乙	淫	音	飮	邑	凝	應
			윤			융	은			을	음			읍	응	
DADX	依	儀	宜	意	疑	矣	義	衣	議	醫	二	以	伊	易	異	·
	의										이					
DAEX	·	移	而	耳	貳	益	翼	人	仁	印	因	姻	寅	引	忍	認
						익		인								
DAFX	一	壹	日	逸	任	壬	妊	賃	入	剩	姉	姿	子	字	慈	·
	일				임				입	잉	자					

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DB4X	・	滋	磁	紫	者	自	資	作	殘	潛	蠶	雜	場	壯	獎	將
		자						작	잔	잠		잡	장			
DB5X	帳	張	狀	章	粧	腸	臟	莊	葬	藏	裝	長	障	再	在	・
														재		
DB6X	・	宰	才	材	栽	災	裁	財	載	爭	低	底	抵	著	貯	寂
										쟁	저					적
DB7X	摘	敵	的	積	籍	績	赤	跡	適	傳	全	典	前	專	展	・
										전						
DB8X	・	戰	殿	田	轉	錢	電	切	節	絶	占	店	漸	點	接	丁
								절			점				접	정
DB9X	井	亭	停	定	庭	廷	情	政	整	晶	正	淨	程	精	訂	・
DBAX	・	貞	鄭	靜	頂	鼎	制	堤	帝	弟	提	濟	祭	第	製	諸
							제									
DBBX	除	際	題	兆	助	措	操	早	曺	朝	條	槽	潮	燥	照	・
				조												
DBCX	・	祖	租	組	調	趙	造	鳥	族	足	存	尊	卒	宗	從	種
									족		존		졸	종		
DBDX	終	綜	鍾	佐	左	座	罪	主	住	周	奏	宙	州	晝	朱	
				좌			죄	주								

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DD4X	·	卓	濁	託	糬	彈	炭	奪	脫	探	塔	湯	兌	台	太	怠
		탁				탄		탈		탐	탑	탕	태			
DD5X	態	泰	胎	宅	擇	澤	土	討	痛	統	通	退	投	透	鬪	·
				택			토		통			퇴	투			
DD6X	·	特	波	派	破	罷	判	板	版	販	八	敗	貝	便	片	篇
		특	파				판				팔	패		편		
DD7X	編	坪	平	評	廢	幣	肺	閉	包	布	抱	捕	浦	胞	幅	·
		평			폐				포						폭	
DD8X	·	暴	爆	標	票	表	品	豊	風	彼	皮	被	避	必	畢	筆
				표			품	풍		피				필		
DD9X	下	何	夏	河	荷	學	鶴	寒	恨	旱	漢	翰	閑	限	韓	·
	하					학		한								
DDAX	·	割	含	咸	艦	合	抗	港	航	降	項	害	海	解	該	骸
		할	함			합	항					해				
DDBX	核	幸	行	向	鄕	響	香	虛	許	憲	獻	軒	險	驗	赫	·
	핵	행		향				허		헌			험		혁	
DDCX	·	革	峴	玄	現	絃	縣	賢	鉉	顯	血	協	享	兄	刑	型
			현								혈	협	형			
DDDX	形	炯	螢	衡	惠	慧	互	呼	好	戶	浩	湖	濠	虎	號	·
					혜		호									

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DC4X	·	柱	株	注	洲	珠	舟	走	酒	週	鑄	駐	竹	俊	準	遵
		주											죽	준		
DC5X	中	仲	衆	重	即	增	曾	症	蒸	證	贈	之	地	志	持	·
	중				즉	증						지				
DC6X	·	指	支	旨	智	枝	止	池	知	祉	紙	脂	至	芝	遲	誌
DC7X	直	織	職	振	晉	津	珍	眞	進	辰	鎭	陣	陳	震	疾	·
	직			진											질	
DC8X	·	秩	質	執	輯	集	徵	且	借	差	次	此	車	着	錯	燦
				집			징	차						착		찬
DC9X	讚	贊	察	札	參	倉	創	唱	彰	昌	暢	窓	蒼	債	彩	·
			찰		참	창								채		
DCAX	·	採	菜	蔡	冊	策	責	妻	處	尺	拓	陟	千	天	川	泉
					책			처		척			천			
DCBX	薦	踐	遷	哲	徹	撤	澈	鐵	尖	添	廳	淸	聽	請	靑	·
				철					첨		청					
DCCX	·	替	諦	體	初	招	礎	秒	草	超	促	燭	觸	寸	村	總
		체			초						촉			촌		총
DCDX	崔	最	推	秋	追	丑	畜	祝	築	縮	蓄	蹴	軸	逐	春	·
	최		추			축									춘	
DCEX	·	出	充	忠	蟲	衝	取	就	醉	側	測	層	値	恥	治	置
		출	충				취			측		층	치			
DCFX	致	齒	則	親	七	漆	侵	枕	浸	針	稱	快	他	妥	打	·
			칙	친	칠		침				칭	쾌	타			

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DE4X	·	護	豪	鎬	惑	婚	昏	混	魂	弘	洪	紅	鴻	化	和	火
		호			혹	혼				홍				화		
DE5X	禾	花	華	畫	話	貨	擴	確	穫	丸	幻	患	換	桓	歡	·
							확			환						
DE6X	·	煥	環	還	活	滑	況	皇	荒	黃	回	悔	懷	會	灰	劃
					활		황				회					획
DE7X	獲	橫	孝	效	曉	侯	候	厚	後	朽	勳	薰	訓	揮	輝	·
		횡	효			후					훈			휘		
DE8X	·	休	凶	胸	黑	欽	吸	興	喜	姬	希	戲	熙	嘉	伽	稼
		휴	흉		흑	흠	흡	흥	희					가		
DE9X	駕	佳	苛	嫁	迦	呵	榎	袈	跏	珏	殼	恪	墾	懇	奸	·
										각			간			
DEAX	·	姦	艱	艮	杆	澗	揀	諫	喝	褐	勘	堪	嵌	戡	淦	瞰
									갈		감					
DEBX	紺	龕	岬	閘	岡	腔	疆	崗	堈	彊	羌	薑	凱	漑	慨	·
			갑		강								개			
DECX	·	价	愷	芥	豈	坑	醵	据	渠	祛	鍵	虔	巾	腱	蹇	杰
						갱	갹	거			건					걸
DEDX	乞	黔	刦	刧	憩	揭	偈	憇	檄	牽	肩	繭	犬	譴	甄	·
		검	겁		게				격	견						
DEEX	·	鵑	鉗	璟	儆	驚	憬	梗	暻	炅	坰	耿	頸	繫	谿	屆
			겸	경										계		
DEFX	誡	詁	枯	膏	錮	拷	雇	痼	賈	攷	哭	梱	控	恭	鞏	·
		고									곡	곤	공			

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
DF4X	·	珙	菓	誇	廓	串	貫	慣	灌	琯	棺	罐	刮	珖	胱	匡
		공	과		곽	관							괄	광		
DF5X	洸	卦	魁	槐	拐	乖	宏	紘	僑	嬌	絞	膠	喬	攪	咬	·
		괘	괴				굉		교							
DF6X	·	狡	皎	蛟	溝	嘔	俱	懼	謳	驅	灸	玖	苟	軀	鳩	鷗
					구											
DF7X	垢	枸	臼	狗	鞠	麴	掘	堀	屈	窟	窮	勸	捲	闕	蹶	·
					국		굴				궁	권		궐		
DF8X	·	獗	潰	机	鬼	珪	揆	糾	叫	赳	逵	戟	剋	槿	筋	僅
			궤		귀	규						극		근		
DF9X	斤	謹	饉	覃	禽	扱	汲	矜	亘	兢	妓	琪	冀	杞	箕	·
					금	급		긍			기					
DFAX	·	嗜	幾	汽	畸	飢	饑	旡	琦	伎	埼	棊	沂	淇	朞	綺
DFBX	羇	羈	耆	錡	騏	驥	喫	奈	拿	拏	喇	梛	煖	捺	捏	·
							끽	나					난	날		

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E04X	·	囊	娘	孃	撚	捻	怒	弩	駑	膿	惱	訥	紐	尼	匿	蛋
		낭		냥	년	념	노			농	뇌	눌	뉴	니	닉	단
E05X	鍛	湍	緞	疸	撻	譚	膽	覃	錟	畓	沓	遝	撞	塘	幢	·
					달	담				답			당			
E06X	·	棠	戴	擡	袋	堵	悼	稻	蹈	搗	濤	萄	塗	禱	賭	逃
			대			도										
E07X	鍍	纛	瀆	篤	禿	頓	焞	棟	桐	憧	董	胴	瞳	痘	兜	·
			독			돈		동						두		
E08X	·	屯	遁	藤	鄧	騰	癩	裸	烙	珞	駱	酪	瀾	爛	欄	欒
		둔		등			라		락				란			
E09X	闌	辣	嵐	濫	藍	襤	拉	臘	萊	掠	亮	輛	樑	倆	粮	·
		랄	람				랍		래	략	량					
E0AX	·	驪	侶	戾	黎	濾	轢	煉	漣	憐	楝	湅	輦	裂	劣	冽
		려					력	련						렬		
E0BX	濂	獵	玲	翎	齡	鈴	囹	岺	怜	醴	隷	撈	虜	蘆	鹿	·
	렴	렵	령							례		로			록	
E0CX	·	碌	弄	籠	聾	賂	儡	雷	療	寥	遼	淚	壘	樓	漏	屢
			롱			뢰			료			루				
E0DX	縷	硫	謬	琉	榴	溜	瘤	戮	淪	侖	崙	綸	慄	溧	勒	·
		류						륙	륜				률		륵	
E0EX	·	肋	菱	凌	綾	裡	裏	浬	痢	罹	麟	燐	潾	淋	琳	霖
			릉			리					린			림		
E0FX	笠	靡	膜	寞	漠	莫	挽	瞞	漫	蠻	娩	巒	蔓	輓	抹	·
	립	마	막				만								말	

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E14X	·	韎	妄	忙	枚	魅	昧	邁	猛	萌	勉	棉	冕	俛	麵	蔑
		말	망		매				맹		면					멸
E15X	冥	暝	螟	酩	耗	侮	冒	慕	牡	矛	摸	牟	姆	帽	摹	·
	명				모											
E16X	·	瑁	穆	沒	歿	蒙	廟	描	昴	猫	撫	誣	毋	巫	拇	畝
			목	몰		몽	묘				무					
E17X	蕪	汶	紊	蚊	勿	瀰	眉	渼	媚	嵋	靡	玟	珉	悶	愍	·
		문			물	미						민				
E18X	·	憫	旼	旻	蜜	駁	泊	薄	剝	拍	撲	粕	樸	珀	縛	潘
					밀	박										반
E19X	伴	搬	攀	磻	斑	頒	畔	絆	魃	勃	撥	渤	跋	妨	尨	·
									발					방		
E1AX	·	彷	肪	倣	膀	坊	幇	昉	榜	謗	龐	盃	陪	賠	俳	胚
												배				
E1BX	栢	魄	柏	飜	樊	煩	藩	蕃	伐	筏	閥	帆	氾	泛	汎	·
	백			번					벌			범				
E1CX	·	梵	范	僻	壁	闢	霹	辨	釆	柄	甁	竝	昞	浜	軿	餠
				벽				변		병						
E1DX	堡	輔	報	褓	馥	僕	宓	袱	輻	蓬	烽	葑	捧	棒	逢	·
	보				복					봉						

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E24X	·	峯	琫	埠	訃	赴	剖	敷	阜	孚	斧	腑	芙	賻	鮒	扮
		봉		부												분
E25X	奮	奔	糞	雰	盆	棚	崩	鵬	痺	匪	卑	沸	翡	匕	婢	·
						붕			비							
E26X	·	毘	琵	秕	脾	臂	蜚	裨	誹	譬	鄙	彬	斌	濱	嬪	瀕
												빈				
E27X	牝	聘	憑	馮	赦	獅	砂	嗣	奢	娑	泗	祀	祠	邪	瀉	·
		빙			사											
E28X	·	似	唆	蛇	巳	篩	紗	麝	賜	朔	傘	撒	薩	杉	衫	插
										삭	산	살				삽
E29X	箱	翔	嘗	峠	庠	甞	賽	嗇	牲	叙	嶼	舒	婿	恕	抒	·
	상						새	색	생	서						
E2AX	·	晢	逝	捿	曙	薯	棲	黍	奭	昔	潟	惜	晳	腺	膳	璿
									석					선		
E2BX	煽	繕	銑	瑄	詵	扇	璇	洩	屑	渫	泄	薛	蟾	殲	閃	·
								설					섬			
E2CX	·	攝	晟	醒	惺	猩	踈	掃	紹	遡	沼	蔬	韶	梳	泝	疏
		섭	성				소									
E2DX	篠	蕭	釗	鮹	贖	粟	遜	淞	悚	淞	誦	碎	鎖	衰	獸	·
					속		손		송			쇄		쇠	수	
E2EX	·	師	搜	穗	繡	蒐	袖	睡	囚	琇	粹	讐	酬	銖	狩	誰
E2FX	羞	隋	髓	叔	旬	循	珣	筍	殉	盾	洵	瞬	荀	蓴	諄	·
				숙	순											

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E34X	·	馴	駒	瑟	拾	濕	繩	丞	陞	屍	侍	柴	矢	媤	蝕	湜
		순		슬	습		승			시					식	
E35X	拭	熄	軾	娠	薪	迅	呻	紳	蜃	訊	尋	瀋	辻	雅	芽	·
				신							심		십	아		
E36X	·	啞	蛾	訝	餓	踆	衙	堊	愕	按	晏	雁	鞍	鴈	斡	軋
								악		안					알	
E37X	庵	菴	闇	狎	秧	鴦	昻	埃	涯	隘	碍	崖	厄	倻	也	·
	암			압	앙			애					액	야		
E38X	·	冶	惹	椰	爺	躍	釀	襄	恙	痒	禦	圄	憶	焉	諺	掩
						약	양				어		억	언		엄
E39X	儼	輿	予	轝	淢	亦	衍	捐	娟	硯	筵	鳶	閱	厭	艶	·
		여			역		연						열	염	엽	
E3AX	·	燁	曄	暎	盈	瑛	嬰	塋	瀛	詠	蘙	叡	枘	洩	曳	猊
				영								예				
E3BX	藝	詣	傲	奧	悟	嗚	墺	吾	旿	伍	鈺	穩	甕	擁	瓮	·
			오								옥	온	옹			
E3CX	·	蝸	訛	莞	琓	腕	玩	頑	浣	阮	婉	宛	曰	枉	汪	歪
		와		완									왈	왕		왜
E3DX	矮	畏	隈	堯	窯	邀	擾	妖	遙	耀	饒	凹	夭	拗	辱	·
		외		요											욕	

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E44X	·	褥	俑	傭	瑢	熔	茸	涌	湧	蓉	佑	虞	又	瑀	寓	尤
		욕	용								우					
E45X	迂	隅	郁	昱	炫	頊	殞	芸	耘	蔚	鬱	媛	寃	洹	垣	·
			욱				운			울		원				
E46X	·	猿	瑗	袁	鴛	尉	萎	魏	逶	謂	渭	瑋	韋	踰	宥	猶
						위								유		
E47X	唯	柚	悠	愉	諭	釉	攸	楢	游	癒	毓	胤	玧	閏	鈗	·
											육	윤				
E48X	·	絨	殷	垠	誾	陰	吟	泣	鷹	膺	誼	毅	擬	椅	懿	痍
		융	은			음		읍	응		의					이
E49X	弛	彝	己	荑	夷	姨	爾	珥	溺	翊	瀷	靭	咽	湮	靷	·
								익				인				
E4AX	·	溢	鎰	馹	姙	叺	込	孕	玆	恣	茨	疵	藉	刺	諮	瓷
		일			임	입		잉	자							
E4BX	仔	柘	煮	昨	酌	雀	棧	盞	岑	暫	璋	丈	仗	匠	杖	·
				작			잔		잠		장					
E4CX	·	掌	蔣	獐	墻	樟	薔	醬	哉	滓	沮	苧	邸	咀	猪	柢
									재		저					
E4DX	楮	蹟	滴	糴	賊	顚	塡	銓	栓	餞	悛	剪	奠	煎	箭	·
		적				전										
E4EX	·	纏	鈿	窃	折	截	浙	点	粘	霑	蝶	蹀	楨	汀	偵	呈
				절				점			접		정			
E4FX	旌	禎	艇	征	挺	町	錠	晸	珽	靖	娗	幀	逞	釘	娣	·
															제	

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E54X	·	齊	劑	悌	蹄	啼	祚	曹	彫	釣	吊	弔	洮	眺	棗	粗
		제					조									
E55X	繰	躁	遭	阻	凋	漕	澡	肇	藻	詛	錞	雕	簇	拙	腫	·
													족	졸	종	
E56X	·	縱	踪	琮	踵	倧	淙	坐	挫	嗾	註	做	呪	疇	胄	厨
								좌		주						
E57X	綢	姝	廚	湊	炷	紬	誅	輳	粥	浚	濬	竣	駿	峻	准	·
									죽	준						
E58X	·	唆	樽	焌	雋	憎	繒	脂	摯	址	趾	只	稙	稷	璡	診
						증		지					직		진	
E59X	秦	塵	疹	盡	軫	榛	瑨	賑	迭	姪	跌	窒	膣	膣	嫉	·
									질							
E5AX	·	帙	朕	斟	什	汁	懲	澄	蹉	遮	叉	捉	搾	贊	纂	瓚
			짐				징		차			착		찬		
E5BX	餐	璨	鑽	饌	撰	刹	擦	斬	站	慘	嶄	瘡	昶	敞	滄	·
						찰		참				창				

7 자료편

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E64X	·	廠	槍	脹	娼	菖	采	寨	凄	悽	隻	戚	拶	斥	滌	瘠
		창					채		처		척					
E65X	脊	喘	淺	闡	擅	阡	喆	轍	凸	綴	籤	帖	疊	諜	捷	·
		천					철				첨	첩				
E66X	·	妾	牒	貼	褺	聽	晴	滯	遞	締	逮	涕	礁	肖	楚	焦
						청		체					초			
E67X	硝	哨	抄	剿	樵	艸	酢	醋	嘱	叢	銃	塚	寵	愡	聡	·
									촉	총						
E68X	·	撮	催	墜	醜	抽	槌	樞	湫	趨	錘	椎	楸	芻	酋	鰍
		촬	최	추												
E69X	蹙	椿	朮	沖	衷	贅	臭	炊	聚	脆	趣	吹	翠	娶	峙	·
	축	춘	출	충		췌	취								치	
E6AX	·	侈	幟	熾	稚	緻	馳	痴	癡	勅	鍼	寢	蟄	陀	墮	惰
										칙	침		칩	타		
E6BX	唾	沱	舵	詫	馱	駝	拓	琸	托	晫	灘	嘆	綻	誕	歎	·
							탁				탄					
E6CX	·	耽	嗿	貪	蕩	宕	颱	跆	苔	殆	汰	邰	吐	兎	堆	頹
		탐			탕		태						토		퇴	
E6DX	褪	套	巴	把	播	婆	芭	跛	坂	阪	辦	覇	牌	膨	澎	·
		투	파						판			패		팽		
E6EX	·	彭	偏	遍	扁	鞭	萍	弊	蔽	斃	陛	暴	砲	鉋	怖	葡
			편				평	폐				포				
E6FX	圃	抛	舖	泡	褒	哺	苞	蒲	飽	瀑	彪	杓	漂	俵	剽	·
										폭	표					

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E74X	·	豹	標	稟	楓	諷	詖	披	疲	疋	弼	泌	匹	珌	乏	逼
		표		품	풍		피			필					핍	
E75X	瑕	賀	廈	蝦	霞	虐	汗	桿	澣	轄	函	啣	陷	陜	蛤	·
	하					학	한			할	함			합		
E76X	·	閤	巷	恒	亢	肛	咳	孩	楷	蟹	諧	邂	劾	杏	享	餉
			항				해						핵	행	향	
E77X	墟	歇	爀	炫	懸	縣	眩	玹	弦	舷	穴	嫌	狹	脅	挾	·
	허	헐	혁	현							혈	혐	협			
E78X	·	俠	夾	浹	脇	馨	桁	熒	珩	荊	彗	蕙	蹊	乎	毫	醐
						형					혜			호		
E79X	昊	糊	壕	澔	皓	胡	頀	弧	晧	壺	扈	滸	瑚	酷	或	·
														혹		
E7AX	·	琿	渾	忽	虹	禍	靴	碻	喚	圜	晥	濶	豁	凰	晃	徨
		혼		홀	홍	화		확	환			활		황		
E7BX	惶	慌	煌	幌	榥	隍	繪	賄	恢	檜	膾	晦	酵	涍	哮	·
							회						효			
E7CX	·	嚆	效	爻	肴	朽	后	喉	嗅	焄	熏	燻	勛	塤	薨	萱
						후				훈					훙	훤
E7DX	諠	毁	徽	暉	彙	携	烋	恤	兇	痕	欣	屹	訖	歆	洽	·
		훼	휘			휴		휼	흉	흔		흘		흠	흡	

7 자료편

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E84X	·	禧	嬉	戱	犧	稀	凞	羲	嘻	詰	哥	柯	笳	侃	癎	竿
		희								힐	가			간		
E85X	憾	柑	匣	慷	糠	愾	箇	喀	騫	怯	鎌	冏	涇	頴	瓊	·
	감		갑	강		개		객	건	겁	겸	경				
E86X	·	痙	竟	莖	鯨	悸	稽	叩	姑	敲	皐	睾	誥	鵠	昆	棍
						계		고						곡	곤	
E87X	拱	龔	瓜	瓘	曠	罫	愧	轟	驕	勾	仇	寇	嶇	柩	毆	·
	공		과	관	광	괘	괴	굉	교	구						
E88X	·	瞿	矩	駒	穹	倦	眷	詭	硅	窺	葵	閨	橘	棘	隙	瑾
					궁	권		궤	규				귤	극		근
E89X	芹	擒	衿	崎	祺	麒	儺	枏	涅	帑	啖	曇	黛	屠	桃	·
		금		기			나	남	녈	노	담		대	도		
E8AX	·	淘	滔	睹	跳	犢	惇	燉	沌	乭	疼	螺	剌	籃	蠟	狼
						독	돈			돌	동	라	랄	람	랍	랑
E8BX	勵	閭	廉	斂	殮	簾	櫓	鹵	麓	牢	寮	燎	瞭	婁	陋	·
	려		렴				로		록	뢰	료			루		
E8CX	·	瑠	凜	稜	璃	躪	鱗	痲	卍	沫	罔	茫	邙	煤	罵	冪
		류	름	릉	리		린	마	만	말	망			매		멱
E8DX	溟	皿	茅	杳	錨	憮	懋	吻	薇	謐	箔	磐	潑	醱	鉢	·
	명		모	묘		무			미	밀	박	반	발			
E8EX	·	徘	輩	帛	癖	瞥	並	屛	菩	俯	傅	孵	鼻	噴	忿	焚
		배		백	벽	별	병		보	부				분		
E8FX	彿	硼	庇	扉	斐	枇	沘	砒	緋	殯	些	徙	捨	裟	珊	
	불	붕	비							빈	사				산	

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E94X		蒜	霰	滲	澁	湘	爽	璽	穡	甥	犀	胥	鋤	鼠	羨	偰
		산		삼	삽	상		새	색	생	서				선	설
E95X	聶	塑	宵	逍	邵	巽	嫂	戍	晬	瘦	竪	雖	塾	夙	楯	·
	섭	소				손	수						숙		순	
E96X	·	脣	醇	嵩	膝	匙	柿	猜	諡	埴	寔	晨	燼	悉	沁	芯
				숭	슬	시				식		신		실	심	
E97X	俄	娥	謁	闕	岩	艾	櫻	鶯	耶	臆	堰	繹	姸	涓	冉	·
	아		알		암	애	앵		야	억	언	역	연		염	
E98X	·	焰	琰	閻	咏	甯	瑩	瓔	穎	刈	睿	芮	裔	敖	雍	壅
					영					예					옹	
E99X	邕	渦	蛙	皖	僥	瑤	慵	盂	雩	彧	円	冤	沅	轅	葦	·
		와		완	요		용	우		욱	원				위	
E9AX	·	蹂	臾	庾	堉	戎	蔭	揖	蟻	怡	翌	刃	荏	咨	炙	雌
		유			육	융	음	읍	의	이	익	인	임	자		
E9BX	勺	灼	爵	鵲	箴	庄	檣	牆	梓	齋	錚	儲	姐	狙	箸	·
	작				잠	장			재		쟁	저				
E9CX	·	這	嫡	磧	笛	翟	迪	迹	廛	箋	詮	渟	碇	穽	酊	鉦
			적						전			정				
E9DX	梯	嘲	爪	稠	糟	詔	猝	瑽	鐘	咫	叱	佚	嵯	懺	倡	·
	제	조					졸	종		지	질		차	참	창	

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
EA4X	·	猖	埰	砦	柵	剔	擲	穿	賤	諂	梢	蕉	蜀	忖	怱	葱
		창	채		책	척		천		첨	초		촉	촌	총	
EA5X	錐	錘	雛	竺	虫	仄	痔	輜	雉	稱	楕	喙	擢	濯	琢	·
	추			축	충	측	치			칭	타	탁				
EA6X	·	呑	坦	眈	搭	撐	桶	筒	腿	妬	坡	琶	頗	叭	唄	悖
		탄		탐	탑	탱	통		퇴	투	파			팔	패	
EA7X	狽	稗	烹	咆	疱	袍	脯	豐	昰	謔	瀚	喊	涵	緘	沆	·
			팽	포				풍	하	학	한	함			항	
EA8X	·	亥	偕	垓	倖	饗	絢	衒	孑	頁	瀅	邢	嵇	鞋	狐	縞
		해			행	향	현		혈		형		혜		호	
EA9X	蝴	顥	惚	哄	泓	訌	奐	宦	轘	鐶	猾	恍	徊	淮	獪	·
			홀	홍			환				활	황	회			
EAAX	·	誨	吼	逅	喙	麾	匈	吃	恰	熹	熺					
			후		훼	휘	흉	흘	흡	희						